

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-242573

(P2000-242573A)

(43) 公開日 平成12年9月8日 (2000.9.8)

(51) Int.Cl.⁷

G 0 6 F 13/00

識別記号

3 0 1

F I

G 0 6 F 13/00

テーマコード (参考)

3 0 1 T

審査請求 有 請求項の数37 O L (全 35 頁)

(21) 出願番号 特願2000-39822 (P2000-39822)

(22) 出願日 平成12年2月17日 (2000.2.17)

(31) 優先権主張番号 09/251032

(32) 優先日 平成11年2月18日 (1999.2.18)

(33) 優先権主張国 米国 (U S)

(71) 出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーション

INTERNATIONAL BUSIN
ESS MASCHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(74) 代理人 100086243

弁理士 坂口 博 (外1名)

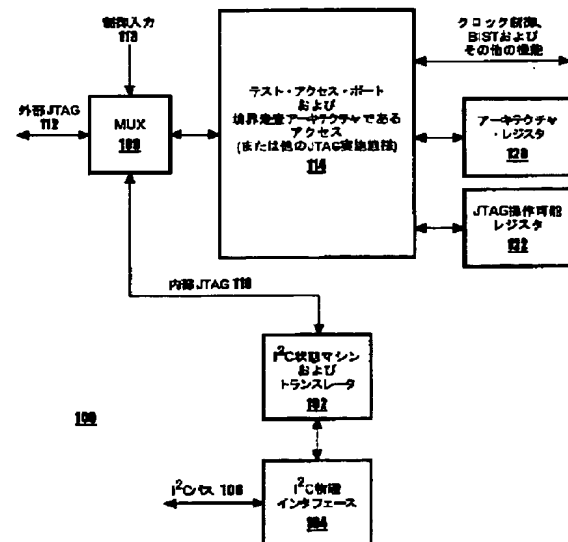
最終頁に続く

(54) 【発明の名称】 I²Cバスを含むグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法および装置

(57) 【要約】

【課題】 集積回路間 (I²C) バスなどのグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法および装置を提供すること。

【解決手段】 入出力バッファ論理回路は、グローバル・シリアル・バスとの間で転送すべきデータをバッファする。入出力バッファ論理回路に接続されたスレーブ・インタフェース論理回路は、データを受信し、そのデータを入出力バッファ論理回路に送信する。入出力バッファ論理回路およびスレーブ・インタフェース論理回路に結合されたスレーブ・コントローラは、入出力バッファ論理回路に合わせてデータ交換のペースを調整する。エラー検出論理回路は、エラー条件を検出するために入出力バッファとグローバル・シリアル・バスとの間に結合される。



【特許請求の範囲】

【請求項 1】グローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置であって、前記グローバル・シリアル・バスとの間で転送すべきデータをバッファする入出力バッファ論理回路と、前記入出力バッファ論理回路に接続され、データを受信し、そのデータを前記入出力バッファ論理回路に送信するスレーブ・インタフェース論理回路と、前記入出力バッファ論理回路および前記スレーブ・インタフェース論理回路に結合され、前記入出力バッファ論理回路に合わせてデータ交換のペースを調整するスレーブ・コントローラを含む装置。

【請求項 2】前記入出力バッファと前記グローバル・シリアル・バスとの間に結合され、エラー条件を処理するエラー処理論理回路をさらに含む、請求項 1 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 3】前記エラー処理論理回路が巡回冗長検査（CRC）計算論理回路を含む、請求項 2 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 4】前記グローバル・シリアル・バスが集積回路間（I²C）バスを含む、請求項 1 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 5】前記製造テスト・インタフェースがジョイント・テスト・アクション・グループ（JTAG）インタフェースを含み、前記スレーブ・インタフェース論理回路が前記 JTAG インタフェースとの間で送受信される JTAG フォーマット・コマンドを処理する、請求項 1 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 6】前記入出力バッファと前記グローバル・シリアル・バスとの間に結合され、エラー条件を処理するエラー処理論理回路をさらに含む、前記 JTAG インタフェースが前記エラー処理論理回路にアテンション信号を供給し、前記アテンション信号がエラーを示す、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 7】前記エラー処理論理回路が巡回冗長検査（CRC）計算論理回路を含み、前記 CRC 計算論理回路が CRC 計算を実現するために使用可能になる、請求項 6 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 8】前記スレーブ・インタフェース論理回路が、前記アテンション信号に応答してエラー検査を使用可能および使用不能にするように構成される、請求項 6 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 9】前記スレーブ・コントローラが、前記スレ

ーブ・インタフェース論理回路と前記 JTAG インタフェースの間のデータ交換のペースを調整する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 10】前記スレーブ・コントローラが、前記スレーブ・インタフェース論理回路と前記 JTAG インタフェースの間のデータ交換のためにデータ・ステアリングおよび経路指定を行う、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 11】前記スレーブ・コントローラが、レジスタ読取り／書込み動作のために順序づけた一連のコマンドを前記スレーブ・インタフェース論理回路に供給する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 12】前記スレーブ・インタフェース論理回路が同時に発生した JTAG 読取り／書込み動作を処理し、前記スレーブ・コントローラが書込み動作に続いて JTAG 命令レジスタ（IR）状況読取り動作のための読取り動作を行う、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 13】前記スレーブ・コントローラが、開始バイト・アドレスに続いてレジスタ読取り／書込み動作のためのレジスタ読取り／書込みアドレスを供給する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 14】前記スレーブ・コントローラが、走査通信レジスタ読取り／書込み動作のために順序づけた一連のコマンドを前記スレーブ・インタフェース論理回路に供給する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 15】前記グローバル・シリアル・バスが、集積回路間（I²C）バスと、前記 I²C バスと前記入出力バッファ論理回路の間に結合された物理インタフェースとを含み、前記スレーブ・コントローラが、前記物理インタフェースと前記入出力バッファ論理回路の間のデータ交換のペースを調整する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 16】前記スレーブ・コントローラが、前記物理インタフェースと前記入出力バッファ論理回路の間のデータ交換のためにデータ・ステアリングを行う、請求項 15 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 17】前記スレーブ・コントローラが、レジスタ読取り動作のために前記スレーブ・インタフェース論理回路にコマンドを供給する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを

接続するための装置。

【請求項 18】前記スレーブ・コントローラが、レジスタ書き込み動作のために前記スレーブ・インタフェース論理回路にコマンドを供給する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 19】前記スレーブ・インタフェース論理回路がテスト・クロック (TCK) 生成論理回路を含む、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 20】前記スレーブ・インタフェース論理回路が前記 JTAG インタフェースからロード・コマンド信号を受信する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 21】前記 JTAG インタフェースと前記スレーブ・インタフェース論理回路が内部 JTAG バスによって接続され、前記内部 JTAG バス上の活動を開始するために前記ロード・コマンド信号が活動化される、請求項 20 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 22】前記グローバル・シリアル・バスが集積回路間 (I²C) バスを含み、エラー処理のために肯定応答信号が使用される、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 23】前記スレーブ・コントローラが、1 つまたは複数の基本動作を実行するために前記スレーブ・インタフェース論理回路にコマンドを供給する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 24】前記基本動作が、前記 JTAG インタフェースへの制御、命令、データ、状況交換のうちの少なくとも 1 つを選択する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 25】前記 JTAG インタフェースが走査通信即時読取り論理回路を含む、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 26】前記走査通信即時読取り論理回路が、ターゲット・レジスタ内容を取り出して走査通信データ・レジスタにロードし、前記走査通信データ・レジスタの前記内容を直ちに読み出すために 1 つのステップを実行する、請求項 5 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

【請求項 27】前記スレーブ・コントローラが、前記入出力バッファ論理回路とのデータ交換のためにデータ・ステアリングおよび経路指定を行う、請求項 1 に記載のグローバル・シリアル・バスに製造テスト・インタフェ

ースを接続するための装置。

【請求項 28】グローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法であって、物理インタフェースと内部バスとの間にスレーブ状態マシンを設け、前記物理インタフェースが前記グローバル・シリアル・バスに接続され、前記内部バスが前記製造テスト・インタフェースに接続されるステップと、前記スレーブ状態マシンを使用するステップであって、前記グローバル・シリアル・バスとの間で転送すべきデータをバッファするステップと、前記製造テスト・インタフェースと前記グローバル・シリアル・バスとの間のデータ交換のペースを調整するステップとを実行するためのステップとを含む方法。

【請求項 29】前記スレーブ状態マシンが、エラー条件を処理するステップをさらに実行する、請求項 28 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

【請求項 30】前記スレーブ状態マシンが、走査通信読取り動作のためのコマンドを供給するステップをさらに実行する、請求項 28 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

【請求項 31】前記スレーブ状態マシンが、走査通信書き込み動作のためのコマンドを供給するステップをさらに実行する、請求項 28 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

【請求項 32】前記スレーブ状態マシンが、制御読取り動作のためのコマンドを供給するステップをさらに実行する、請求項 28 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

【請求項 33】前記スレーブ状態マシンが、制御書き込み動作のためのコマンドを供給するステップをさらに実行する、請求項 28 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

【請求項 34】前記スレーブ状態マシンが、基本書き込み動作のためのコマンドを供給するステップをさらに実行する、請求項 28 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

【請求項 35】前記スレーブ状態マシンが、基本読取り動作のためのコマンドを供給するステップをさらに実行する、請求項 28 に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

【請求項 36】グローバル・シリアル・バスに製造テスト・インタフェースを接続するための集積回路デバイスであって、前記集積回路デバイスが、前記グローバル・シリアル・バスとの間で転送すべきデータをバッファする入出力バッファ論理回路と、前記入出力バッファ論理回路に接続され、データを受信し、そのデータを前記入出力バッファ論理回路に送信す

るスレーブ・インタフェース論理回路と、前記入出力バッファ論理回路および前記スレーブ・インタフェース論理回路に結合され、前記入出力バッファ論理回路に合わせてデータ交換のペースを調整するスレーブ・コントローラとを含む集積回路デバイス。

【請求項 37】前記入出力バッファ論理回路と前記スレーブ・インタフェース論理回路の間に結合されたエラー処理論理回路をさらに含む、請求項 36 に記載の集積回路デバイス。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般にデータ処理分野に関し、より具体的には集積回路間（I²C）バスなどのグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法および装置に関する。

【0002】

【従来の技術】I²Cバスは、1つまたは複数のマスタおよびスレーブにより複数の集積回路を相互接続するために通常使用する業界標準のシリアル双方向2線式バスである。I²Cバスは、2本のバス・ラインと、1本のシリアル・データ・ライン（SDA）と、1本のシリアル・クロック・ライン（SCA）とを含む。I²Cバスは、nチャネル金属酸化膜半導体（NMOS）、相補型金属酸化膜半導体（CMOS）、バイポーラなど、どのような集積回路（IC）作成プロセスもサポートするものである。シリアル・データ（SDA）およびシリアル・クロック（SLC）情報は、I²Cバスに接続された複数のデバイス間の2本のワイヤによって伝達される。

【0003】集積回路間（I²C）バスを含むグローバル・シリアル・バスにジョイント・テスト・アクション・グループ（JTAG）インタフェースのIEEE1149.1仕様などの製造テスト・インタフェースを接続するための方法および効果的なメカニズムの必要性が存在する。VPD情報を照会し、レジスタ・アクセスを可能にし、メモリおよびIOサブシステムを含み、シリアル・バスに接続された複数のデバイスでテストを実行するためにI²Cバスなどのグローバル・シリアル・バスを使用することは望ましいことである。

【0004】

【発明が解決しようとする課題】本発明の主な目的は、集積回路間（I²C）バスを含むグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法および装置を提供することにある。他の目的は、実質的に否定的な影響がなく、従来技術の配置の欠点の多くを克服するような方法および装置を提供することにある。

【0005】

【課題を解決するための手段】簡単に言えば、集積回路間（I²C）バスなどのグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法およ

び装置が提供される。入出力バッファ論理回路は、グローバル・シリアル・バスとの間で転送すべきデータをバッファする。入出力バッファ論理回路に接続されたスレーブ・インタフェース論理回路は、データを受信し、そのデータを入出力バッファ論理回路に送信する。入出力バッファ論理回路およびスレーブ・インタフェース論理回路に結合されたスレーブ・コントローラは、入出力バッファ論理回路に合わせてデータ交換のペースを調整する。

10 【0006】本発明の特徴によれば、エラーを検出するために入出力バッファとグローバル・シリアル・バスとの間にエラー検出論理回路が結合される。

【0007】

【発明の実施の形態】次に図面を参照すると、図1は、参照文字100によって一般的に示されている本発明による製造テスト・インタフェースおよびグローバル集積回路間（I²C）バス相互接続装置を示している。製造テスト・インタフェースおよびグローバル集積回路間（I²C）バス相互接続装置100は、I²C物理インタフェース104を介してI²Cバス106に接続されたスレーブI²C状態マシンおよびトランスレータ102を含む。製造テスト・インタフェースおよびI²Cバス相互接続装置100は単一集積回路チップによって定義することができる。I²C状態マシンおよびトランスレータ102は、内部JTAGバス110を介してマルチプレクサ（MUX）108に接続されている。MUX108は外部JTAGバス112に接続されている。制御入力113は、内部JTAGバス110と外部JTAGバス112との選択を行うためにMUX108に印加される。テスト・アクセス・ポートおよび境界走査アーキテクチャであるアクセス114または他のJTAG実施態様は、MUX108と、クロック制御、組込み自己試験（BIST）およびその他の機能118、最高2³³個のアーキテクチャ・レジスタ120、JTAG走査可能レジスタまたはテスト・データ・レジスタ（TDR）122との間に接続される。

【0008】好ましい実施の形態の特徴によれば、製造テスト・インタフェースおよびI²Cバス相互接続装置100は、固有のやり方でI²Cバス106とJTAGテスト・アクセス・ポート（TAP）アクセス114との間のギャップを橋絡し、相互接続装置が固有のI²Cレジスタ転送プロトコルと完全なJTAGの機能性の両方をサポートできるようにする。製造テスト・インタフェースおよびI²Cバス相互接続装置100は、業界標準のI²Cバス・プロトコルを固守するが、JTAGの機能性を制限するものではない。製造テスト・インタフェースおよびI²Cバス相互接続装置100は、相互接続配線を最小限にし、チップレベルのアドレス可能性を取り入れ、直接メモリ・マップ入出力（MMIO）読取りおよび書込みレジスタ・アドレス可能性を定義する。

【0009】好ましい実施の形態の特徴によれば、製造テスト・インタフェースおよびI²Cバス相互接続装置100は、回路内データ経路、テスト、および研究室／現場デバッグ機能に製造テスト・ハードウェアを使用できるようにする。また、製造テスト・インタフェースおよびI²Cバス相互接続装置100は、新しい走査コントローラ命令である走査通信即時読取りを取り入れ、I²Cバス106上のレジスタ読取り応答待ち時間を短縮する。製造テスト・インタフェースおよびI²Cバス相互接続装置100は、システムレベルの製造テスト用の接続を簡略化し、削減する。システム・レベルの製造テストでは、テスト・インタフェース用として単一のI²C接続106を使用し、統合テストおよびバーンイン・テストを実行するために必要なテスト・フックアップ手順およびソフトウェアを簡略化することができる。

【0010】I²C状態マシンおよびトランスレータ102は、I²Cバス106により送信されたデータを固有のJTAGコマンドに変換する。I²C状態マシンおよびトランスレータ102は、I²Cバス106に接続されたチップ内でレジスタの読取り、書込み、走査を実行するためにアクセス114とのインタフェースをとるTAPコントローラとして使用することができる。I²C状態マシンおよびトランスレータ102は、論理回路組込み自己試験(LBIST)またはアレイ組込み自己試験(ABIST)を開始し、JTAG TDR122を走査し、チップ・クロックならびにJTAGバス112を介して開始することができ、アクセス114によって実現される他の機能を制御することができる。製造テスト・インタフェースおよびI²Cバス相互接続装置100は7ビットのスレーブ・アドレス指定方式を使用し、2⁷個のスレーブを直接アドレス指定することができる。このプロトコルは、特定のスレーブによって定義されるメモリ内の場所から始め、連続バイトをシリアルにアドレス指定することにより、アドレス指定されたスレーブが読取り／書込みに応答することを暗示する。製造テスト・インタフェースおよびI²Cバス相互接続装置100は、直接アドレス指定を2²³まで拡張し、8バイト整合のレジスタを操作する。さらに、レジスタ読取り／書込み以外にアクセス114によってサポートされる動作が用意されている。

【0011】シリアルI²Cバス106は、バスが休止している間、1のままになる、シリアル・データ(SDA)用とシリアル・クロック(SDC)用の2本のワイヤからなる。どのようなバス・マスタも、開始ビット(SCL=「1」である間、SDAが1→0になる)によってメッセージ転送を開始することができる。I²Cバス・プロトコルは、それぞれが1バイト幅で、後に肯定応答ビット(0=良好な肯定応答)が続くパケットを操作する。メッセージが始まると、スレーブは肯定応答ビット・タイミング中にSDAの駆動を支配し、マス

タはそれ以外の時期にSCLとSDAの駆動を支配する。マスタ論理回路は必ずクロックを支配する。スレーブは、肯定応答ビットを除き、読取り時にSDAを支配する。SDAがSCLの立上り前にセットアップされ、立下り後に保持され、SCLがローである間に遷移するように、各SDAデータ／肯定応答ビットは、SCLがクロックする(0→1→0)間に1または0の安定状態を維持しなければならない。

【0012】パケット・ビットは、最上位ビット(ビット1)から先に伝送される。開始バイトのビット1:7は特定のスレーブ・チップをアドレス指定する。開始バイトのビット8は読取り／書込みビットである。書込みは、マスタのバイト送信およびスレーブの肯定応答とともに進行する。マスタが開始バイトを書き込むと読取りが始まるが、スレーブが開始バイトを肯定応答した後、データフローは反転する。次にスレーブはパケットを送信し、マスタは肯定応答する。マスタがAck=1になるまで、スレーブは送信し続ける。すべての伝送は停止ビット(SDAが0→1になる間、SCL=1)で終了する。バスは、停止ビット後、もう一度休止する。

【0013】IEEE1149.1では、境界走査アーキテクチャとやりとりするためのテスト・アクセス・ポートと呼ばれる5線インタフェースを定義している。アクセス114は、ジョイント・テスト・アクション・グループ(JTAG)インタフェースのIEEE1149.1仕様の具体的な実施態様である。JTAGテスト・アクセス・ポート(TAP)の5本のワイヤは、JTAGテスト・クロック入力(TCK)と、JTAGテスト・データ入力(TDI)、JTAGテスト・データ出力(TDO)と、JTAGテスト・モード選択入力(TMS)と、JTAGテスト論理リセット入力(TRST)とを含む。テスト・クロック入力(TCK)の立上りにより、TMSおよびTDIはアクセス114によってサンプリングされる。

【0014】TCKの立上り時のJTAGテスト・モード選択入力(TMS)の値により、アクセス114のTAPコントローラにおいて状態遷移が行われる。テスト・データ入力(TDI)は、アクセス114へのシリアル・データ入力である。テスト・データ出力(TDO)は、アクセス114からのシリアル・データ出力である。JTAGテスト論理リセット入力(TRST)により、アクセス114を回復するためにテスト論理の非同期リセットが行われる(TAP→Test Logic Reset)。Shift IRは、アクセス114内のTDIとTDOの間で命令がシリアルにシフトされる状態である。TMSが0に保持されている間、各TCKクロックにより、もう1つのビットがTDIから命令レジスタ内にシフトインされ、IR状況がTDOからシフトアウトされる。命令レジスタの内容により、アドレス指定される具体的なTDRまたは実行すべき非レジスタ動作が

決定される。ShiftDRは、多くのデータ・レジスタ（TDR）の1つがTDIとTDOの間にシリアルに接続される状態である。TMSが0に保持されている間、各TCKクロックにより、もう1つのビットがTDIから命令レジスタ内にシフトインされ、（古い）データがTDOからシフトアウトされる。

【0015】図2は、一般的に200で示されている好ましい実施の形態によるI²C状態マシンおよびトランスレータ102の1次スレーブ・データおよび制御レジスタを示している。I²Cスレーブ論理回路200は、データ、アドレス、レジスタ読取り／書込み（R/W）バッファのための入出力バッファ・レジスタ（IOBUF）201と、関連の入出力バイト・カウント・レジスタ（IOCNT）202とを含む。I²Cスレーブ論理回路200は、物理インタフェース104とスレーブJTAGインタフェース204の間に結合されたエラー検出203を含む。また、I²Cスレーブ論理回路200は、物理インタフェース104、入出力バイト・カウント・レジスタ（IOCNT）202、スレーブJTAGインタフェース204に結合されたスレーブ・コントローラ206を含む。スレーブJTAGインタフェース204は、シリアル・シフト・レジスタ（SSR）208と、そのシリアル・シフト・レジスタ（SSR）208に関連するビット・カウント・レジスタ（BCR）210とを含む。また、スレーブJTAGインタフェース204は、テスト・モード選択（TMS）選択レジスタ（TSR）212およびTMS端子選択レジスタ（TTSR）214と、テスト・クロック入力（TCK）を供給するクロック・ジェネレータ（CG）216と、JTAGテスト・モード選択入力（TMS）を供給するMUX218とを含む。

【0016】物理インタフェース104は、I²Cプロトコルおよびタイミングを処理し、関連のハンドシェイク信号とともに1バイト幅のデータ・インタフェースを提供する。SDAは、I²Cによって定義される双方向シリアル・データ信号を意味する。また、SCLは、I²Cによって定義されるシリアル・クロック信号を意味する。物理インタフェース104の動作は、一般にI²C規格の従来の実施態様に応じて行われる。

【0017】IOバッファ（IOBUF）201は、I²C開始バイトによってスレーブ・チップとしてアドレス指定されるチップとの間で転送される連続バイトをバッファする。IOBUF201内のデータは、物理インタフェース104との間で一度に1バイトずつパラレルに転送される。開始バイト全体はIOBUF201内に保持されるが、読取り／書込みビットはI²C状態マシン102によってさらに使用される。書込み動作中のデータ・フローは、I²Cバス106から内部JTAGバス110のTDIまたはTMSに向かう。読取り動作中のデータ・フローは、内部JTAGバス110のTDO

からI²Cバス106に向かう。シリアル・シフト・レジスタ（SSR）208は、IOBUF201との間で最高8バイトをパラレルに受信または供給することができる。次にこのデータは、IEEE1149.1プロトコルに従う内部JTAGバス110との間でシリアルにシフトされる。

【0018】物理インタフェース104は、I²Cバス106を監視し、着信／発信バイトを受信する。そのバイトが開始バイトである場合、物理インタフェース104は7ビットのスレーブ・アドレスをチップ・アドレスと比較する。アドレスが一致しない場合、開始バイトと残りのメッセージは無視される。物理インタフェース104は、スレーブ・コントローラ206に対して信号をアサートし、1バイトが受信されたことを示す。

【0019】スレーブ・コントローラ206は、IOカウント（IOCNT）202を減分してIOBUF201との間のデータ交換のペースを調整し、データをIOBUF201内で適切に位置合わせして保持する。メッセージの終わりを受信または送信するかあるいはIOBUFがオーバフローまたはアンダーフローすると、スレーブ・コントローラ206は、SSR208とIOBUF201の間でデータを転送し、1つまたは複数のJTAG基本動作を実行するようにスレーブJTAGインタフェース204をトリガする。本明細書および特許請求の範囲で使用する基本動作とは、内部スレーブJTAGインタフェース204との間でテスト・モード選択（TMS）、制御、またはTDI／TDO、命令またはデータ／状況を駆動するための1つまたは複数のテスト・クロック入力（TCK）パルスからなるシーケンスを意味するものと定義されている。

【0020】図3は、IOBUF201およびIOCNT202を実現するスレーブIOバッファ論理回路を示している。スレーブIOバッファ論理回路201は、IOカウント・レジスタ（IOCNTL2）302と、アップ／ダウン・カウンタ304と、比較器（>「1011」）306と、ANDゲート307と、IOバッファ（IOBUFL2）308と、1対のマルチプレクサ（MUX IOA、MUX IOB）310、312とを含む。IOCNT202は、IOカウント・レジスタ（IOCNTL2）302およびアップ／ダウン・カウンタ304と同等のものである。

【0021】休止している場合、IOカウント・レジスタ（IOCNTL2）302には、IOバッファ201の最下位バイトを指し示す、b「1011」というカウント・データ（CNTDATA）がパラレルロードされる。I²C状態マシン102は、IOCNTL2302に印加されたLOADCNTをアサートすることにより、CNTDATAが有効であることを示す。I²Cバス106により複数バイトが転送されるにつれて、カウントは減分され、次の最上位バイトを連続的に指し示

す。I²C物理インタフェース104はビットの転送（最上位ビットが先）を処理し、IOBUF201は、I²C物理インタフェース104に複数バイトを提示し、それから複数バイトを受け入れる（最下位バイトが先）。

【0022】開始バイトがI²Cバス106上での読取り動作を示す場合、I²C状態マシン102は、I²Cバス106にデータを返す準備として、LOADCNTを活動状態としてアサートし、CNTDATA=b「0111」としてアサートすることにより、IOバッファ201のデータ部分までIOCNTをジャンプする。

【0023】IOCNTレジスタIOCNTL2302は、基本動作ステップ・カウンタ304としてレジスタ読取り／書込み動作中に再使用される。このモードでは、レジスタ読取り／書込みシーケンス内の次の基本コマンド・ステップが実行されるたびに、ステップ・カウンタ304に印加されるCountUpがI²C状態マシン102によって活動化される。IOCNTL2と、IOBUF201のデータ部分のみが変更され、IOBUFアドレスおよび開始バイトはそれぞれの値を保持する。

【0024】12バイトのIOバッファ・レジスタ（IOBUFL2）308は、一度に1バイトずつ開始バイトから始めて右から左にロードされる。I²C物理インタフェース104からのデータは、信号IS_BYTE_0上で

到着し、I²C状態マシン102からの信号COUNT BYTEによって有効であると示される。MUX IOA310は、IOCNTによって指し示されるバイトを除くすべてのバイトにIOBUFデータをフィードバックするが、これはIS_BYTE_0上のデータによって置き換えられる。

【0025】読取り情報は、信号IS_BYTE_I上で物理インタフェース104に提示される。MUX IOB312は、IS_BYTE_I上で駆動すべきIOCNT番目のバイトを選択する。BUFLOAD信号がI²C状態マシン102によって活動化されると、SSR208からのデータがIOBUF201のデータ部分に平行にロードされる。これは、レジスタ読取り基本シーケンスの終わりまたはI²Cコマンドの要求で1ステップのJTAG基本コマンドが実行された後に行われる。

【0026】データ、コマンド／アドレス、r/wビットは、IOBUFL2（0:95）信号によりI²C状態マシン102およびSSR208に提示される。ANDゲート307によって生成されるOVERRUN信号は、IOBUFの満杯／空条件を示す。

【0027】以下の表1は、xが無指定を示す場合のマルチプレクサ（MUX IOA、MUX IOB）310、312に関するMUX定義を示すものである。

【0028】

【表1】

表1: MUX定義

MUX IOA310 出力	CountByte	RW	IOcntL2	BufLoad
SSRL2(0:63) & IOBUFL2(64:95)	x	x	x	1
IOBUFL2(0:95)	0	x	x	0
IS_BYTE_O(0:7) & IOBUFL2(8:95)	1	0	0000	0
IOBUFL2(0:7) & IS_BYTE_O(0:7) & IOBUFL2(15:95)	1	0 0	0001 0111	0
etc.	1	x	etc	0
IOBUFL2(0:87) & IS_BYTE_O(0:7)	1	x	1011	0
無指定 (**オーバーラン・エラー、過剰データ受信**)	1	x	>1011	0
MUX IOB312 出力				
IOBUFL2(0:7)	x		x000	x
IOBUFL2(8:15)	x		x001	x
etc	x		etc	x
IOBUFL2(56:63)	x		x111	x

【0029】MUX IOB312はIOcntL2の最上位ビットに関して無指定であり、スレーブ論理回路はデータのみを返し、アドレス情報は一切返さない。

【0030】図4は、JTAGインタフェース204を

実現するスレーブ・テスト・アクセス・ポート（TAP）インタフェース論理回路を示している。スレーブTAPインタフェース論理回路204は、TAPクロック状態マシン（TAPCMSL2）402と、複数のマル

チプレクサ (MUX A、MUX B、MUX C、MUX D、MUX E) 404、406、408、410、412と、複数のレジスタ (SSRL2 (0:63)、TTSRL2、TSRL2、BCRL2 (0:5)) 414、416、418、420と、ダウン・カウンタ421と、比較器 (=「111111」) 422と、複数のラッチ413、424、426と、1対のANDゲート428および430を含む。

【0031】ロード・コマンド (LOADCMD) 信号が活動化されると、テスト・アクセス・ポート・コマンド (TAPCMD) は、内部JTAGバス110上での活動を開始するために有効であると見なされる。TAPCMD信号上の情報は以下に記載するようにコマンド・レジスタ内にロードされ、1クロック (ONECLK2) ラッチ426は1つまたは複数のTCKパルスでリリースするように1に設定される。以下の表2、表3、表4に示すTAPクロック状態マシン (TAPCSM) 402、TAP基本コマンド・インタフェース、MUX D410に関する定義を参照されたい。MUX B406は、フィードバックを行い、LOADCMDが活動

状態ではない間、コマンド・レジスタの現在の値を保持

表2: TAPクロック状態マシン (TAPCSML2) 402

状態 (エンコード)	入力	NextState	出力
IDLE (00)	ONECLKL2 その他	TCKF1 IDLE	
TCKF1 (01)		TCKR0	BCRL2=111111の場合に ONECLKL2をリセットする
TCKR0 (11)		TCKR1	TCK=1 TDOからTDOL1にロードする
TCKR1 (10)		IDLE	TCK=1 BCRCount=not (BCRL2=「111111」) SSRshift=1

【0034】状態遷移中のTCK上のグリッチを防止するために状態マシン402の状態はグレイコード化される。状態テーブルに特に指定がない限り、出力は0になる。

【0035】MUX A404、MUX B406、BCRL2 420に印加されるTAPコマンド (TAPCMD) 信号は9バイト幅である。第1のバイトはJTAG走査論理回路204によって実行される基本コマンドである。TMS選択レジスタ (TSRL2: TAPCMDビット0) 418は、MUX C408がTMSにシフトアウトすべきSSRL2 (0:63) 414からのSSRデータを選択するかまたは一定値を選択するかを決定する。0は、SSRL2414内のデータがTMS制御情報のストリームであることを示す。1は、最初のN-1個のTCKクロック・パルスのTMS上で駆動される静的0値を選択するようMUXCに指示するものであり、次にTTSRL2 416の値はN番目の(端

する。同様にMUX A404はSSR (データ) 値を保持する。

【0032】TAPクロック状態マシン (TAPCSML2) 402は、ONECLKL2によって活動化されるまでアイドル状態のままになる。状態マシン402は、IDLEに戻る前に、状態TCKF1、TCKR0、TCKR1間を順次移行する。マシン402は、ONECLKL2=1である限り、これらの状態間をループし続ける。TCKはTCK立上り (TCKR0、TCKR1) 状態のときに1に駆動され、ベース・スレーブ・クロック周波数を係数4で効果的に分割し、約50%のデューティ・サイクルを提供する。BCRは減分され、SSRデータはTCKR1状態のときに1ビット右にシフトされる。TCKの立上りでTDOをサンプリングするというJTAGの要件は、TCKR0状態のときにTDOからTDOL2ラッチ413にロードすることによって達成される。以下の表4に示すMUX E412の定義を参照されたい。

【0033】

【表2】

子) TCKパルス中にTMS上で駆動される。

【0036】端子TMS選択レジスタ (TTSRL2: TAPCMDビット1) 416は、N番目の(端子) TCKパルス中にTMS上で値を提供する。TTSRL2 416は、SSRの64ビット幅が走査を完了するには不十分な場合の長時間走査動作のものである。TTSRL2=0は基本動作を橋絡し、他の基本コマンドで走査を続行できるようにする。TTSRL2=1は、アクセス114内のJTAG TAP状態をシフトDRまたはシフトIR状態から移行させることにより、その動作を終了する。テスト論理リセット (TRST) は、本来は基本コマンドに使用されない組合せでTAPCMDビット0:1をゲートすることによって形成される。ANDゲート428のこの出力はラッチ424によってラッチされ、テスト論理リセットTRST信号をデグリッチする。ビット・カウント・レジスタ420 (BCRL2: TAPCMDビット2:7) は、TMSまたはTD

I上の情報シーケンスとともに送信するTCKパルスの数を定義する。b「111110」というBCR値は、64個のTCKパルスを送信することを示す(SSRレジスタ414の全体はTDIまたはTMSから右にシフトされ、TDOを左側のSSR内にシフトインする)。BCR=b「111111」は端子カウントであり、1

表3: TAP基本コマンド・インタフェース

ビット	信号	説明
0	TSRL2 418	TMS 選択レジスタ 0: TMS<=(SSR(63)、右にシフト)* (BCRL2+2) 1: TMS<=0 BCRL2=111111 である間 TTSL2 BCRL2=111111 である場合
1	TTSRL2 416	TMS 端子選択レジスタ TSRL2=1 である場合に BCRL2 カウントの最後のビット中に TMS 上で駆動される値
	TRST	TRST<=(TSRL2=0) および (TTSRL2=1) これは、それ以上のビットを使用せずに TRST 機能を獲得する方法である
2:7	BCRL2 420 (6 ビット)	ビット・カウント・レジスタ シフトするビット数から2を引いた数をロードする: b「111110」=64個のTCKパルス (SSRは63回シフトする) b「111101」=63個のTCKパルス (SSRは62回シフトする) など b「111111」=1つのTCKパルス (SSRはシフトしない) 注: パルス数が0の場合はエンコードを行わない。
8:71	SSRL2 (64 ビット)	単独走査レジスタ TDI からシフトアウトすべきデータ (TSRL2=1 である場合はTMSからもシフトアウトする) TDO->(0)SSRL2(63)->TDI/TMS TDOはTCKの立上りで取り込まれ、 TDI/TMSはTCKの立下りでランチされる (SSRがシフトされる)

【0039】TAPCMD信号の残りのビットは、TDIまたはTMSからのシフトアウトの準備としてSSR208内にロードすべきデータを表している。

【0040】BCRがb「111111」に達すると、そのコマンドは実行される。ONECLKL2ラッチ426は0に戻り、TAPCMD状態マシン402はアイドル状態のままになる。内部JTAGバス110は非活動状態になる。ANDゲート430のCMDDONE信号出力は、新しい基本コマンドを開始してもよいという

つのTCKパルスがリリースされることを示す。

【0037】TAPCMDの第1のバイトの定義を以下の表3に要約して示す。

【0038】

【表3】

インジケータとしてI²Cスレーブ状態マシン102にフィードバックされる。

【0041】マルチプレクサ (MUX A、MUX B、MUX C、MUX D、MUX E) 404、406、408、410、412に関するMUX定義を示す以下の表4では、xは無指定を示し、/=は等しくないことを意味する。

【0042】

【表4】

表 4 : MUX 定義

MUX A (0:63) 404 出力	Load Cmd	SSR shift	TSRL2	BCRL2	TAPcmdSM
SSRL2 (0:63)	0	0	x	x	x
TDO & SSRL2 (0:62)	0	1	x	x	x
TAPCMD (8:71)	1	x	x	x	x
MUX B (0:1) 406 出力					
TSRL2 & TTSRL2	0	x	x	x	x
TAPCMD (0:1)	1	x	x	x	x
MUX C 408 出力					
SSRL2 (63)	x	x	0	x	x
0	x	x	1	/=111111	x
TTSRL2	x	x	1	=111111	x
MUX D 410 出力					
1	1	x	x	x	x
ONECLKL2	0	x	x	x	/=TCKF1
ONECLKL2	0	x	x	/=111111	TCKF1
0	0	x	x	=111111	TCKF1
MUX E 412 出力					
TDO	x	x	x	x	TCKR0
TDOL2	x	x	x	x	/=TCKR0

【0043】図5は、スレーブ・コントローラ206を実現するスレーブ・コントローラ論理回路を示している。スレーブ・コントローラ206は、I²Cスレーブ状態マシン論理回路(I²CSSML2(0:3))502と、比較器(=)504と、TAP CMD経路指定用のMUX SMA506、データ経路指定用のMUX SMB508、制御用のMUX SMD510という複数のマルチプレクサを含む。また、スレーブ・コントローラ論理回路206は、レジスタ読取り/書込み(REGRW)バス同期論理回路512を含む。REGRWバス同期論理回路512は、1対のANDゲート514、516と、SCアイドル・ラッチ(SCIDLEL2)520と、NOTゲート522と、マルチプレクサMUXSMB524を含む。スレーブ・コントローラ論理回路206は複数の基本機能を以下のように実行する。I²Cスレーブ状態マシン論理回路(I²CSSML2(0:3))502は、物理インタフェース106とIOバッファ論理回路300の間のデータ交換のペースを調整し、IOバッファ論理回路300とJTAG走査論理回路204の間のデータ交換のペースを調整する。TAP CMD経路指定用のMUX SMA506は、レジスタ読取り/書込み動作の順序づけた一連のコマンドを提供する。REGRWバス同期論理回路512は、JTAG走査論理回路204とアクセス114の間のデータ交換のペースを調整する。データ経路指定用

のMUX SMB508は、データ経路指定用のI²C帯域幅要件を最小限にする。

【0044】表5、表6、表7、表8は、スレーブ・コントローラ206のMUX SMA506、MUX SMB524、MUX SMC508、MUX SMD510に関するマルチプレクサ定義を示している。

【0045】IOBUF201およびSSR208データは、必ずバイトごとに転送するわけではない。I²Cが基本コマンド・シーケンスを使用してIOBUF/S SRレジスタの幅の偶数倍より短いかまたは等しくないTDRをアドレス指定する場合、すべての重要データを受信したときにオーバーフローを検出できるように、I²Cバス106からのデータはIOBUF201内で左揃えされる。IOBUF201データは、SSR208にロードされると右揃えされる。ターゲットになるTDRから走査されたデータは、SSR208内で左揃えされ、同じく適切な時期にI²Cに返されたデータのアンダーフローを検出できるように、IOBUF201に転送されたときに左揃えされたままになる。

【0046】REGRWバス同期論理回路512は、レジスタ読取り/書込みバス同期のためにSCidle信号を使用する。CHKADR状態の場合、SCidleL2ラッチ520は1に設定される(アクセス114からのSCIDLE信号は活動状態であると想定する)。レジスタR/W動作が開始されると、SCIDLEはロ

一になり、ScidLeL2520をリセットする。レジスタ読取り動作が完了すると、SCIDLEL2ではなくSCIDLEが、レジスタ読取りデータがアクセス114内で使用可能であることを示す。この論理回路を使用してレジスタ読取りバスと同期をとるために、第3のステップの完了後にTABLE内のレジスタ読取りシーケンスが休止される。

【0047】図6は、IOBUF201と物理インタフェース104の間のデータ転送のペースを調整するために必要な信号を示している。信号定義については、状態

10
表5: MUX 定義

MUX SMA 506 出力	CmdMux Sel	コメント
IOBUFL2 (80:87) & TapDataBits (0:63)	0000	TAP コマンド
		レジスタ読取り 基本 TAP コマンド・シーケンス:
x'08 00000000 000000DF	0001	1. TAP=home, shiftir
x'DE 00000000 14' & IOBUFL2 (64:87)	0010	2. IR=registerRDimmed, SSR=status, TAP=exitir
x'00 00000000 00000001	0011	3. TAP=RunTestIdle (RegisterRWDONE の場合、ここで休止する)
x'01 00000000 00000001	0100	4. TAP=shiftdr
x'FE00000000 00000000	0101	5. SSR=scscan (0:63), TAP=exitdr
x'00 00000000 00000001	0110	6. TMS=UpdateDR, Idle
x'00 00000000 00000000	0111	7. ノーオペレーション (TMS=0、TCK=2つのクロック)、アイドル
無指定	1000	未実現

【表6】

表5の続き

		レジスタ書き込み基本 TAP コマンド・シーケンス:
x'08 00000000 000000DF	1001	1. TAP=home, shiftir
x'DE 00000000 0F 000040	1010	2. IR=scan, SSR=status, TAP=exitir
x'02 00000000 00000003	1011	3. TAP=shiftDR
x'FE' & IOBUFL2 (0:63)	1100	4. scscan (0:63)=SSR TAP=exitdr
x'08 00000000 000000DF	1101	5. TAP=home, shiftir
x'DE 00000000 18' & IOBUFL2 (64:87)	1110	6. IR=registerwtm SSR=status, TAP=exitir
x'00 00000000 00000001	1111	7. Tap=RunTestIdle (トリガ書き込み)

マシンの説明を参照すると最もよく理解できるだろう。

【0048】図7は、IOバッファ201とJTAG走査論理回路204とのデータ・ペーシングを示している。図7は、IOBUF201とJTAG走査論理回路204の間のデータ転送のペースを調整するために使用する信号を示している。信号定義については、状態マシンの説明を参照すると最もよく理解できるだろう。

【0049】

【表5】

表6: MUX 定義

MUX SMB 出力	I2CSSML2	コメント
1	=ChkAdr	
Scidle および SCidleL2	その他	

【0051】

【表8】

表7: MUX 定義

MUX SMC 出力	IOBUFL2	コメント
ValidBytes=「0000」 である場合、TapDataBits= 「00000000000000」X お よび IOBUFL2 (0:7) にな り、ValidBytes=「0001」 である場合、TapDataBits= 「00000000000000」X およ び IOBUFL2 (0:15)などに なる。 ValidBytes 出力: $1 \leq (BCR+2) \leq 8$ である場合は b「0000」 $9 \leq (BCR+2) \leq 15$ など である場合は b「0001」 $57 \leq (BCR+2) \leq 64$ である場合は b「0111」	基本アドレス	TapDataBits 出力: 64 ビット・フィールドが 以下のように作成される: IOBUFL2 の有効デー タ・ バイト (IOBUFL2(82:87) によって決定される)は、 右揃えされ、定数「00」X で左側が埋められて、64 ビット・フィールド(8 バ イト)を作成する。 ValidBytes 出力: 基本コマンドの伝送サイ ズに応じて「000」から 「111」までの値(すなわ ち、IOBUFL2 内の有効デ ータの LSByte を指し示 すポインタとして使用さ れる IOBUFL2(82:87)に よる)
TapDataBits: IOBUFL2(0-63) ValidBytes: 「111」	その他	基本コマンドではない場 合、TapDataBits は使用し ない。非凸本コマンドの 場合、LSByte は必ず 「111」になる(すべて 8 バイトの IOBUF データが 使用される)。

【0052】

【表9】

表 8

MUX SMD 出力	I2CSSML2	IOCNTL2	IOBUFL2	コメント
b「0001」	ChkAdr	x	x	IOCNTL2 は、この時点ではレジスタ読取り／書込みシーケンス内のステップをカウントするためのアップ・カウンタとして使用する。
ValidBytes	ChkStop WrtAck	x b「0111」	primitive addr primitive addr	IOBUFL2 内で有効な LSByte の位置を IOCNTL2 にロードする。
b「1011」	その他			12C Addr バイトへの充填を開始するために IOBUFL2 の右端バイトを指し示す。

【0053】次に図8を参照すると、エラー検出203を実現するエラー処理論理回路が示されている。エラー処理論理回路203は、I²Cバス106との間で転送されたデータのCRC検査を行うための巡回冗長検査(CRC)論理回路802を含む。また、エラー処理論理回路203は、CRC_Good、アテンション、IOCNTL2、read_or_write、比較器808からの基本コマンドという入力を受け取るACK TYPE MUX804を含む。

【0054】I²C状態マシン102とエラー処理論理回路203は、JTAG TAPコントローラ・アクセス114からアテンション入力を受け取る。アクセス114からのこのアテンション入力は、レジスタ読取り／書込みバス・エラーの結果としてまたはいくつかの他のエラー・シナリオにより活動化される場合がある。read_or_writeは、物理インタフェース104からバッファされたI²C状態マシンからの入力である。IOBUFL2は、IOバッファ論理回路201から比較器808への入力である。比較器808に印加されるTapCmdAdrは、内部定数としてまたはチップIOピンを介して構成可能な定数である。IOCNTL2は、IOバッファ論理回路201からMUX804への入力である。CRC_Goodは、CRCバイトが正しく転送されたことを示す、CRC論理回路802からの信号である。is_slave_ack_typeは、エラー処理論理回路800のMUX804から物理インタフェース104への出力である。

【0055】I²Cスレーブは肯定応答パルスによってエラーを処理する。I²Cでは、Ackまたは9番目のクロック・サイクルならびにスレーブ・アドレスの9番目のビット上のデータのすべてのバイトについて肯定応答が必要である。スレーブがエラーを検出すると、スレーブは、読取りシーケンスを実行しようと試みる時に

もはやそのアドレスおよびデータを肯定応答しなくなる。このようにして、ソフトウェアは、エラー条件が存在すると判定し、I²Cバス106を使用して、そのエラー条件からの回復を行えるように何がそのエラーを引き起こしたかを判定することができる。

【0056】読取りまたは書込みシーケンス時にスレーブが肯定応答しなくなる状況はいくつか考えられる。たとえば、CRC検査が使用可能になっている場合、ソフトウェアがデータを上書きするのを防止するために、スレーブは13番目のバイトを肯定応答しなくなる。このような13バイトとしては、スレーブ・アドレスと、3バイトのレジスタ読取りアドレスと、8つのデータ・バイトと、1つのCRCバイトとを含む。CRCが正しく計算され、伝送された場合のみ、書込みが行われる。CRC検査を行わない場合、8バイトを上回る量のデータをソフトウェアが書き込むと、データは上書きされる。アクセス114内でアテンション信号が立ち上がり、そのアテンションが内部でブロックされていない場合、状態マシンは、レジスタ読取り直接シーケンスを実行しようと試みる時にNo-Ackを引き起こすことになる。書込みシーケンスを実行し、送信された3バイト・アドレスが基本アドレスではなかった場合、スレーブは、I²Cバス106により送信されるデータ・バイトを肯定応答しなくなる。CRC検査が使用可能になっており、I²Cバス106上で送信されたCRCが正しくなかった場合、スレーブは次の読取り試行時に肯定応答しなくなる。

【0057】

【表10】

表 9: エラー処理論理回路の ACK TYPE MUX804 定義

is_slave_ack_type	Attention, read_or_write, Primitive Cmd disable_attention	コメント
0	Primitive Cmd=1 Attention=x read_or_write=x	内部 IOBUFL2 (64~79) が TapCmdAdr を指し示すときは必ず肯定応答
0	Primitive Cmd=0 read_or_write=0 およびバイト 11、10、9 を指し示す IOCNTL2 ま たは disable_attention=1	アドレス・バイトを書き込むかまたはアテンションが使用不能になっているときは肯定応答
ATTENTION	Primitive Cmd=0 read_or_write=1 または read_or_write=0 およびバイトを指し示す IOCNTL2<=8	読取り時または IOBUFL2 内に非基本アドレスを含むデータ・バイトを書き込むときは ATTENTION 入力に基づいて否定/肯定応答

【0058】次に図 9、図 10、図 11、図 12、図 13、図 14、図 15 を参照すると、図 9 には状態マシン 102 の順次状態が示されている。図 10 および図 11 は、I²C スレーブ状態マシン 102 の基本書込み制御フローおよび基本読取り制御フローをそれぞれ示している。図 12 および図 13 は、I²C スレーブ状態マシン 102 のレジスタ書込み制御フローおよびレジスタ読取り制御フローをそれぞれ示している。図 14 および図 15 は、I²C スレーブ状態マシン 102 の制御書込み制御フローおよび制御読取り制御フローをそれぞれ示している。

【0059】I²C スレーブ状態マシン 102 の IDLE 状態 902 では I²C 活動が一切行われない。IOCNTL2 には b「1011」がロードされ、開始バイトの受信準備として IOBUFL2 の LSByte を指し示す。複数の状態としては AMATCH 状態 904 と、WBYTE 状態 906 と、CHKOVRFLW 状態 908 と、WRTACK 状態 910 とを含む I²C 書込み状態がある。書込み I²C 状態では、I²C マスタがスレーブに複数バイトを書き込んでいるときに、第 1 のバイトは必ず、スレーブをアドレス指定する開始バイトになる。開始バイトの 8 番目のビットは読取り/書込みビットである。次の 3 バイトは、レジスタ・アドレス、基本 TAP コマンド、またはスレーブ制御コマンドのいずれかとして解釈される。残りのバイトはデータである。

【0060】AMATCH 状態 904 (アドレス一致) では、物理インタフェース 104 は開始ビットを認識済みであり、(次の)メッセージ・バイトをアセンブルしている。AMATCH 状態 904 は、反復開始であるかまたは開始バイト・アドレスが不一致であるときに IDLE 状態 902 に戻り、スレーブを IDLE に送り返す。最初の 4 バイト (開始+アドレス・バイト) を受信する前にメッセージが終了した場合、スレーブは IDLE

E に戻る。CRC が使用可能になり、CRC エラーが検出された場合、AMATCH 状態 904 は IDLE 状態 902 に戻る。AMATCH 状態 904 は、物理インタフェース 104 が 8 ビット以上を受信したときに、WBYTE 状態 906 に移行する。物理インタフェース 104 のある特定の実施態様は、開始バイト中に IsStartDetected の前に Is8times を示し、それを紛失しないように Is8times をラッチすることを要求する (Is8timesL2)。AMATCH 状態 904 は、1 バイト以上のデータを受信したあとで物理インタフェース 104 が停止ビットを検出したときに、CHKADR 状態 912 に移行する。

【0061】WBYTE 状態 906 (バイト書込み) では、物理インタフェースからの IsByte_0 が IOBUFL2 に書き込まれ (IOCNTL2th byte)、次に IOCNTL2 が減分される。WBYTE 状態 906 は、これが I²C 書込み動作である場合に、CHKOVRFLW 状態 908 に移行する。データフローはマスタ/イニシエータからスレーブへと継続する。WBYTE 状態 906 は、これが I²C 読取り動作であるときに、CHKADR 状態 912 に移行する。データフローは開始バイト後に反転する。次にマスタはスレーブからデータを受信することになる。

【0062】CHKOVRFLW 状態 908 (IO バッファ内のオーバ (アンダー) フローの検査) では、いかなるアクションも行われない。CHKOVRFLW 状態 908 は、これが書込み動作であり、IOCNTL2 が b「0000」から b「1111」に折り返しておらず、IOBUFL2 内に追加バイト用のスペースがあるときに、WRTACK 状態 910 に移行する。CHKOVRFLW 状態 908 は、レジスタ読取りまたは基本コマンドがすでに IOBUFL2 を充填し、I²C 読取り状態のうちの RBYTE 状態 932 から CHKOVRF

LW状態908に入った場合に、WAITREAD状態910に移行する。1つまたはそれ以上のバイトはすでに読み取られているが、IOBUFL2はまだアンダーフローしていない。CHKADRに移行し、IOBUFL2は満杯/空であり、レジスタ書き込み/読取りまたは基本動作をトリガする。

【0063】WRTACK状態910（スレーブに書き込まれたばかりのI²Cバイトの肯定応答）は、最初の4バイト分のメッセージ（開始+3つのアドレス・バイト）内である場合に良好肯定応答アクションを提供する。WRTACK状態910は、レジスタ読取りアドレスおよびアテンションがアクセス114から活動状態になっている場合に肯定応答失敗を提供する。IsDoRead（I²Cマスタによって書き込まれたバイトのIOBUF読取り）は物理インタフェース104に対して活動化され、IOBUF用に1バイトが用意されていることを示す。NeedAckL2はリセットされる。これが4番目のバイト（IOBUFに書き込まれたばかりの最後のアドレス・バイト：この時点でIOcntL2=7）であり、アドレスが基本コマンドを示す場合、データ・バッファ内でIOcntL2をジャンプするために基本のBCRフィールドからデコードされるValidBytesを調べる。たとえば、BCR=b「00000」である場合、2ビットだけがアクセス114に伝送されるので、1バイト分のデータだけが必要になる。第1のバイトのデータの直後にオーバーフローが発生するように、IOcntL2はb「0000」まで進む。それ以外の場合は、I²Cは、さらに7バイト分の重要ではないデータを伝送して時間を浪費しなければならないだろう。

【0064】WRTACK状態910はNeedAckL2によってAMATCH状態904に移行する。物理インタフェース104は、ackが送信/受信された後の1サイクルの間、IsAckSentを活動化する。Ackが送信されるまでスレーブ状態マシンが待機していた場合、NeedAckL2は記憶している。

【0065】CHKADR状態912（アドレス検査）では、IOBUFL2アドレス・フィールドを調べて、これがI²Cスレーブ・コマンド・フォーマットを使用するレジスタ読取り/書き込み、基本、または制御コマンドのいずれであるかを判定する。基本信号は、基本動作と制御動作の両方について活動状態になる。ScidleL2は、これがレジスタ読取り/書き込み動作である場合に設定される。REGRWバス同期論理回路512はレジスタ読取り/書き込み（アーキテクチャ・レジスタr/w）バスを監視して、JTAGコマンド・ステップがレジスタ読取り/書き込み動作をオーバーランしないようにする。IOcntL2レジスタは、これがレジスタ読取り/書き込み動作である場合にb「0001」に設定される。この場合、IOCNTは、レジスタ読取り/書き込み

動作を遂行するために基本TAPコマンドの定義済みシーケンスを進行する。

【0066】CHKADR状態912はIDLE状態902に移行し、アテンションが活動状態であるかまたはCRCが使用可能になっていて比較が一致しない場合、レジスタ読取りは肯定応答に失敗することになる。CHKADR状態912は、IOBUFL2アドレスが基本ベース・アドレスと一致しないときに、レジスタ読取り/書き込みコマンド状態、REGRWGO状態918のいずれか一方に移行する。また、CHKADR状態912は、IOBUFL2アドレスが基本r/wコマンド・アドレスと一致したときに、基本コマンド状態、LDTAPCMD状態914のいずれか一方に移行する。CHKADR状態912は、IOBUFL2アドレスが制御コマンド・アドレスの1つと一致したときに、CHKSTOP状態924に移行する。これらのコマンドはJTAGバス活動がなくても遂行される。

【0067】基本コマンド状態としては、LDTAPCMD状態914と、WAITTAPDONE状態916とを含む。IOBUFL2アドレス・バイトは、オンチップJTAGインタフェースに直接通信するために予約されたアドレスと一致する。TMSまたはTDI/TDO活動のいずれか一方が行われるが、両方が行われるわけではない。IOBUFL2のバイト10にロードされた基本TAPコマンドに応じて、SSRL2レジスタの幅までの任意の数のビットが転送される。

【0068】LDTAPCMD状態914（TAPコマンドのロード）では、TAPコマンドがIOBUFL2（バイト10）からJTAGインタフェース204にロードされる。LoadCmd信号とValidBytes信号は一緒に機能して、SSRL2の右端バイト内に重要なバイト整列データをロードする。これは、データ・フィールド内で右揃えされたデータをI²Cが送信しなければならないことを暗示する。たとえば、BCR=b「000111」である場合、9ビットがアクセス114に伝送され、I²C物理インタフェース104からのデータのLSByteは8つの有効ビットを含まなければならない。I²C物理インタフェース104からの次のバイトは1つの有効ビットを含み、そのビットはそのバイト内の右端のビットとして送信しなければならない。

【0069】WAITTAPDONE状態916（TAP完了の待機）では、いかなるアクションも行われなない。JTAGインタフェース204は活発に基本TAPコマンドを実行している。WAITTAPDONE状態916は、JTAGインタフェース204からのCmdDoneによってCHKSTOP状態924に移行し、ハードウェアが終了した時期を通知する。

【0070】CHKSTOP状態924（I²C停止条件の検査）では、レジスタ読取り/書き込み動作は必ず8

バイトを転送するが、基本コマンドはデータの一部のみがSSRL2で有効になって終了する可能性がある。IOcntL2 302には、いくつかの左揃えバイトが有効データを含むかを示すValidBytesがロードされる。たとえば、BCR210はb「000111」から開始した(9ビット転送)。SSRL2は初めに右揃えした9ビットを保持したので、バイト6、7のみが有効であった。基本コマンドが完了した後、TDOデータはSSRL2 414内に左揃えで取り込まれたので、バイト0、1のみが有効である。

【0071】CHKSTOP状態924は、I²C停止ビットが検出されたときにIDLE状態902に移行し、追加のアクションは一切不要である。これは、レジスタ読み取り/書き込み動作が長いタイムアウトで遅延しない限り、ありそうもない経路である。また、CHKSTOP状態924は、I²C物理インタフェース104に返す必要がある読み取りデータがSSR208内に存在するときに、LOADRESULTS状態926に移行する。さらに、CHKSTOP状態924は、I²C物理インタフェース104によって要求された書き込みが完了し、スレーブが転送された最新バイトを肯定応答(ACK)するのをI²C物理インタフェース104が期待しているときに、WRTACK状態910に移行する。

【0072】I²C読み取り状態としては、LOADRESULTS状態926と、READACK状態928と、WAITREAD状態930と、RBYTE状態932とを含む。

【0073】I²C読み取り状態では、I²C物理インタフェース104がスレーブから複数バイトを読み取っている。レジスタ読み取り、基本、またはLOADRESULTS状態926およびREADACK状態928が続く制御読み取りは、IOバッファ201がデータを使い果たすたびに実行される。マスタは、スレーブから戻すすべてのタイミングを支配する。マスタは、1つまたは複数のバイトを読み取り、Ack=1であるときにI²C転送を終了することができる。WAITREAD状態930と、RBYTE状態932と、CHKOVRFLW状態908によって形成されるループは、転送されるすべてのバイトごとに1回実行される。

【0074】LOADRESULTS状態926(SSRからIOBUFへのデータのロード)では、実行されたばかりのJTAG動作が基本である場合、データは依然としてSSRL2 414内にあり、IOBUFL2 308にコピーしなければならない。それ以外の場合、これは後述するレジスタ読み取り/書き込み動作のステップ6ですでに実行されており、SSRは不要情報である。

【0075】READACK状態928(スレーブから読み取ったばかりのI²Cバイトの肯定応答の待機)では、物理インタフェース104は、Ackビットの待機

を含む、読み取りタイミングを管理する。図6も参照すると、物理インタフェース104がAckを受信した後、物理インタフェース104は、NeedAckL2ラッチを設定するIAckReadyを活動化することにより、それがIOBUF201からさらに1バイトを必要とすることを示す。IsWrtReadyは、物理インタフェース104に戻って活動化され、次のバイトがIOBUF201からIs_Byte_Iに書き込む用意ができていることを示す。同時に、NeedAckL2はリセットされる。

【0076】READACK状態928は、物理インタフェース104が活発に他のデータ・バイトをマスタに返送しているか、またはAck=1の場合に停止ビットを待っており、それが読み取りの終了時にI²Cプロトコルによって許される唯一可能な条件である場合に、WAITREAD状態930に移行する。

【0077】WAITREAD状態930(I²Cによるスレーブからのバイト読み取りの待機)では、いかなるアクションも行われない。物理インタフェース104は、活発に他のデータ・バイトをマスタに返送しているか、またはAck=1の場合に停止ビットを待っている。

【0078】WAITREAD状態930は、マスタによって読み取られた最初のデータ・バイトが「1」で肯定応答され、物理インタフェース104によって停止ビットが検出されると直ちにIDLEへの退出が行われるときに、IDLE状態902に移行する。それ以外の場合、NeedAckL2は、物理インタフェース104がマスタへの他のバイトの送信を完了し、Ack=1を受信し、その際にIDLE状態902への退出が行われる時期を示す。IDLE状態902はIsWrtReadyを活動化し、物理インタフェース104が停止ビットの検出を開始し、NeedAckL2をリセットし、停止ビットを消費できるようにする。WAITREAD状態930は、物理インタフェース104がAck=0を受信し、他のデータ・バイトが必要であることを示したときに、RBYTE状態932に移行する。

【0079】RBYTE状態932(他のデータ・バイトを読み取るためのセットアップ)では、IOcntL2 302が減分され、IOBUFL2 308内の次のデータ・バイトを指し示す。

【0080】レジスタ読み取り/書き込み(REGRW)コマンド状態としては、REGRWGO状態918と、WAITCMDDONE状態920と、CMDSTEP状態922とを含む。IOBUFL2アドレス・バイトは、基本または制御コマンド用に予約されたとのアドレスとも一致しない。そのアドレスはレジスタ読み取り/書き込みアドレスとして直接解釈される。TMSとTDI/TDOの両方の活動は、レジスタ読み取り/書き込み動作を遂行するために一連の定義済みステップで行われる。正

確に64ビットであるSSRL2レジスタ414の幅が転送されることになる。

【0081】REGRWGO状態918（レジスタ・コマンド・シーケンスの（次の）ステップの開始）では、レジスタ・シーケンスの（次の）ステップがTapCmdバス上に置かれ、LoadCmdが活動化されてJTAGインタフェース204を開始する。IOcntL2＝ステップ6である場合、レジスタ読取りによって読み取られたデータはSSRL2 414内にあるので、アクセス114のTAPコントローラをRunTestIdleに戻すクリーンアップ・ステップの前にIOBUFL2 308に保管する必要がある、最終コマンド・ステップは本来はSSR内のデータを破壊する恐れがある。これは、BufLoadを活動化することによって遂行される。

【0082】WAITCMDDONE状態920（TAPおよび／またはレジスタ読取り完了の待機）では、いかなるアクションも行われない。JTAG走査論理回路が活発に基本TAPコマンドを実行しているかまたはレジスタ読取り動作が開始されたがまだ完了していない。

【0083】WAITCMDDONE状態920は、最新コマンド・ステップが完了したことをJTAG走査論理回路からのCmdDone信号が示し、アクセスからのSCIDLE信号が非活動状態になり、続いて活動状態への復帰が行われた後でRegisterRWDone信号が活動状態になったときに、CMDSTEP状態922に移行する。RegisterRWDone信号はレジスタ読取りのステップ3の間のみ重要であり、その直後にレジスタ読取り動作が行われるはずである。

【0084】CMDSTEP状態922（次のコマンド・ステップへの前進）では、IOcntL2が増分され、IOcntL2はレジスタ読取り／書き込みコマンド・ステップ全体にインデックスを付けるためにCmdMuxSelで使用される。

【0085】CMDSTEP状態922はSCOMGOに移行し、レジスタ読取りおよび書き込みコマンド・シーケンスはどちらも正確に7ステップの長さである。ステップ7がまだ完了していない場合、次のステップを開始するためのループバックが設けられている。CMDSTEP状態922は、7番目のステップが完了したときに、CHKSTOP状態924に移行する。

【0086】図16は、製造テスト・インタフェースおよびグローバル集積回路間（I²C）バス相互接続装置100のテスト・アクセス・ポート（TAP）および境界走査アーキテクチャであるアクセス114の主要機能を示している。アクセスは、JTAG走査機能1102と走査通信機能1104を提供する。JTAG走査機能1102により、I²Cスレーブ状態マシン102は線TAP/SPに示すテスト・アクセス・ポート（TAP）インタフェース（IEEE1149.1）によりチ

ップ・リングを走査することができる。アテンション出力とパワーオンリセット（POR）出力も線TAP/SPに示す位置に供給される。アテンション出力は、観測または介入／回復を必要とする条件に応じてI²Cスレーブ状態マシン102に警告するために供給される。

【0087】走査通信機能1104は、チップの機能論理回路と走査機能との間のパラレル・データ経路を提供する。アクセス動作は、チップ・クロックが動作し、システムが動作している間に実行することができ、走査通信機能1104はこれを利用して、動作中のチップまたはシステムとの通信を可能にする。

【0088】クロック・ツリー1114は、アクセス114からTCK SCAN CLOCKS入力を受信する。クロック・ツリー1114はチップ・クロックも出力する。任意選択でセミカスタムのLBIST/SCANインタフェース1116を使用して、チップ走査リングをレベル依存形走査設計（LSSD）リングおよびBISTチャネルなどの機能グループに編成する。インタフェース1116は、アクセス114のJTAG走査1102と機能チップ論理回路1118との間に結合される。

【0089】好ましい実施の形態の新しい命令である走査通信即時読取りは、アクセス114のJTAG走査コントローラの設計に追加されたものであり、I²CとJTAGの間の新しいインタフェース機能の設計を容易にするものである。

【0090】新しい命令である走査通信即時読取りの設計以前は、チップ・レジスタの読取りは2つのステップを必要とした。すなわち、ステップ1は、アクセス走査コントローラ・コマンドである「走査通信読取り」（16進コード「17」）を実行して、ターゲット・チップ・レジスタの内容を取り出し、アクセス走査通信データ・レジスタ1104内に内容をロードすることである。ステップ2は、アクセス走査コントローラ・コマンドである「スキャンアウト」（16進コード「12」）を実行して、アクセス走査通信データ・レジスタ1104の内容を読み出すことである。この2ステップ・プロセスは、既存のソフトウェアおよびハードウェアに対応するために設計されたものである。

【0091】この新しい走査通信即時読取りコマンドにより、ステップ1および2は単一ステップに結合される。すなわち、アクセス走査コントローラ・コマンドである「走査通信即時読取り」（16進コード「14」）を実行して、ターゲット・チップ・レジスタ1120の内容を取り出し、アクセス走査通信データ・レジスタ1104内に内容をロードし、アクセス走査通信データ・レジスタ1104の内容を直ちに読み出すことである。

【0092】I²CとJTAGの間の新しいインタフェース機能が元の2ステップ・プロセスを使用しなければならぬ場合、1クロック・サイクル（それ自体のI²

10

20

30

40

50

Cクロック・サイクルのうちの1つ)で2つのステップを実行しようと試みる必要があるだろう。I²Cクロック・サイクルは他のシステム・クロックに関して変動する可能性があるので、両方のステップが正常に完了すると保証することはできない。プロセスを1つのステップに削減することにより、1クロック・サイクルで動作が完了すると保証することができる。

【0093】相互接続装置100およびI²C状態マシン102の動作については、以下の例により理解できるだろう。

【0094】64ビット・レジスタ読取り／書込み動作は開始バイト・アドレス(I²Cスレーブ・ベース・アドレス)から始まり、任意選択で最高3バイトのレジスタ読取り／書込みアドレス(LSByte、上位ビットが先)が続く。一実施の形態では、8バイト整列のレジスタ読取り／書込みアドレスを定義する。書込みはすべて3バイトのレジスタ読取り／書込みアドレスから始めなければならない、読取りは前に書き込まれたアドレスを使用するかまたは再始動／読取りバイトの前にそのアドレスのうちの1バイト、2バイト、または3バイトすべて(CRC検査が使用可能になっていない場合)を変更することができる。I²Cスレーブは、24ビット・レジスタ読取り／書込みアドレス(23から0まで、0=LSB)を実現する。アクセス/JTAGは、LSBをパリティ・ビットとして定義する。残りの23ビット(23から1まで)はバイト整列アドレスを表す。レジスタ読取り／書込み転送は必ず8バイト幅で行われるが、各チップは通常、必ず8バイト境界上に整列することができない可変幅のレジスタを構築する。I²Cレジスタ読取り／書込みデータ・フィールド内のデータの具体的な位置合わせは、チップの実施態様に依存する。停止ビットをこの時点で受信した場合は、書き込まれたアドレスの一部分のみが変更されることになり、他の活動は一切開始されない。

【0095】書込み動作の場合、4番目のバイト(LSByte、上位ビットが先)から開始し、書込み動作用のデータが続く。停止またはオーバフロー条件が1つまたは複数のデータ・バイトに続くと、レジスタ読取り／書込み動作がキックオフされ、レジスタ読取り／書込みアドレスは自動増分式ではない。8バイトを上回るデータが書き込まれる／読み取られると、オーバフロー／アンダーフローが発生する。この設計のオプションとして、8バイトを上回る書込み／読取り用の連続レジスタ読取り／書込みアドレスを書き込む／読み取るためにレジスタ読取り／書込みアドレスを自動増分式にすることができる。8バイトを上回るデータを書き込むと、同じレジスタ読取り／書込みアドレスにデータが上書きされることになる。8バイト未満のデータを書き込むと、完全な8バイトでレジスタ読取り／書込みアドレスが書き込まれることになり、データの最上位バイトは前の動作

からレジスタ読取り／書込みデータIOバッファ内に残っているものとともに書き込まれることになる。CRC検査が使用可能になっている場合、良好なデータ転送に上書きするのを防止するために、スレーブは書込み中に9番目(CRC)のバイトを肯定応答しなくなる。

【0096】読取り動作の場合、I²Cインタフェース上で再始動(停止後に開始が続く)が出され、R/Wビット=「1」になる。これは、I²Cハードウェア内でキックオフすべきレジスタ読取りをトリガするものである。次にデータフローは反転し、スレーブは各肯定応答ごとに1つのデータ・バイトを返す(LSByteが先)。レジスタ読取り／書込みアドレスは自動増分式ではない。8バイトを上回る量が読み取られた場合、そのデータはLSByteから繰り返されることになる。

【0097】非レジスタTAPコマンド・フォーマットは予約されたレジスタ読取り／書込みアドレス(23:12)=x「524」(このアドレスは再配置可能である)によって識別される。TapCmdビット10:8の値によって選択される8通りの動作モードが存在する。ビット11は今後の使用のために予約されている。基本TAPコマンド(TapCmdアドレス(11:8)=「0000」)フォーマットでは、レジスタ読取り／書込みアドレスのLSByteは、TMSまたはTDI上のTAPポートにビットをシフトアウトさせ、TDOを取り込むTAPコマンドとして解釈される。これにより、I²Cインタフェース104はJTAG論理回路によってサポートされるどのアクションでも実行することができる。停止またはオーバフロー条件は、書込み動作およびオーバフロー条件に関するその現行値によりBCRをキックオフする。R/Wアドレス・ビットが1に設定された停止または開始は、読取り動作に関するその現行値によりBCRをキックオフする。「BCR_value mod 8」(残りが非ゼロの場合は+1)バイトを上回るデータを書き込むかまたは読み取ると、BCR再ロードがデータ・ストリームを続行する(すなわち、走査リング内の連続バイトを書き込む／読み取る)。CRC検査が使用可能になっている場合、CRCバイトが正しく伝送されない限り、すべての読取り／書込み基本およびレジスタ読取り／書込みは抑制される。

【0098】Tapコマンド例: 64ビット+ビット・リング・レジスタ読取り／書込み: (CRC使用不能)

内部走査リングの基本64ビット・スキャンアウトに続いて1ビット・スキャンアウトを実行するために必要なI²Cコマンドのシーケンスについて以下に詳述する。

【0099】1. 基本コマンドのロード: Tap状態マシンをまずテスト論理リセットにし、次にシフトIRにして、命令をスキャンインするようスレーブに命令する。

I²C書込み開始w/R/Wビット=0、長さ=5バイ

ト、STOP

バイト1: 08x: 10個のTCKパルス (アドレス/コマンド・バイト後のデータがTMSシーケンスになる)

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

バイト4: DFx: 値のTMSシーケンス (右端のLSBからMSBへ)

バイト5: 00x: 連続した値のTMSシーケンス (Tap SMをシフトIRにする)

【0100】2. 基本コマンドおよび命令データのロード: Tapコントローラ命令レジスタに0F 80 00 41をスキャンインするようスレーブに命令する。これは内部リングのスキャンアウトであり、次にTap S/MをExit1-IRにする。

I²C書込み開始w/R/Wビット=0、長さ=7バイト、STOP

バイト1: DEx: 32個のTCKパルス

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

バイト4: 41x: 命令のLSB

バイト5: 00x: 命令の次のバイト

バイト6: 80x: 命令の次のバイト

バイト7: 0Fx: 命令のMSB (内部走査リングのスキャンアウト)

【0101】3. 基本コマンドのロード: Tap状態マシンを(Exit1-IRから)シフトDRに順序づけるようスレーブに命令する。

I²C書込み開始w/R/Wビット=0、長さ=4バイト、STOP

バイト1: 02x: 4個のTCKパルス

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

バイト4: 03x: 値のTMSシーケンス (右端のLSBからMSBへ、Tap S/MをシフトDRにする)

【0102】4. 基本コマンドのロード: すべてについてTMS=0の64個のTCKパルスを開始し、Tap S/MをシフトDRに保持するようスレーブに命令する。

I²C書込み開始コマンド: すべてについてTMS=0の64個のTCKパルスを開始し、Tap S/MをシフトDRに保持するようスレーブに命令する。

バイト1: BEx: 64個のTCKパルス

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

【0103】5. スキャンアウト・コマンド用の戻りデータの獲得: TMSが0であり、Tap S/MをシフトDRに残す。

I²C読取り再始動w/R/Wビット=1、長さ=8バイト、STOP

バイト1: FEx: データのLSB

バイト2: 0Fx: データの次のバイト

バイト3: DCx: データの次のバイト

バイト4: BAx: データの次のバイト

バイト5: EFx: データの次のバイト

バイト6: BEx: データの次のバイト

バイト7: ADx: データの次のバイト

バイト8: DEx: データのMSBバイト

【0104】6. 次の基本コマンドのロード: TMS=1の1個のTCKパルスを開始し、1ビットSCAN後にTap S/MをExit1-DRにキックするようスレーブに命令する。

I²C書込み開始w/R/Wビット=0、長さ=3バイト、STOP

バイト1: FFx: 1個のTCKパルス

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

【0105】7. スキャンアウト・コマンド用の戻りデータの獲得 (1ビット): Tap S/MをExit1-DRにキックする。

I²C読取り再始動w/R/Wビット=1、長さ=1バイト、STOP

バイト1: FEx: データのLSB (右端のビットのみ有効である)

【0106】8. 次の基本コマンドのロード: Tap状態マシンをテスト論理リセット状態にするようスレーブに命令する。

I²C書込み開始w/R/Wビット=0、長さ=4バイト、STOP

バイト1: 03x: 5個のTCKパルス

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

バイト4: 1Fx: 右端のビットから左端までのTMS値のシーケンス (Tap状態マシンをリセットに戻すために5ビットのみ使用する)

【0107】Tapコマンド例: 2. 5バイト・レジスタ書込み: 内部走査リングに2. 5バイト (20ビット) のデータをスキャンインするために必要なI²Cコマンドのシーケンスについて以下に詳述する (注: 結果として得られるスキャンイン・データは16進のEF BAになる)。

【0108】1. 基本コマンドのロード: Tap状態マシンをまずテスト論理リセットにし、次にシフトIRにして、命令をスキャンインするようスレーブに命令する。

I²C書込み開始w/R/Wビット=0、長さ=5バイト、STOP

バイト1: 08x: 10個のTCKパルス (アドレス/コマンド・バイト後のデータがTMSシーケンスになる)

バイト2: 40x: 基本コマンド・アドレスのLSB
 バイト3: 52x: 基本コマンド・アドレスのMSB
 バイト4: Dfx: 値のTMSシーケンス (右端のLSBからMSBへ)

バイト5: 00x: 連続した値のTMSシーケンス (Tap SMをシフトIRにする)

【0109】2. 基本コマンドおよび命令データのロード: Tapコントローラ命令レジスタに0F 80 00 41をスキャンインするようスレーブに命令する。これは内部リングのスキャンアウトであり、次にTap S/MをExit1-IRにする。

I²C書込み開始w/R/Wビット=0、長さ=7バイト、STOP

バイト1: DEX: 32個のTCKパルス

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

バイト4: 41x: 命令のLSB

バイト5: 00x: 命令の次のバイト

バイト6: 80x: 命令の次のバイト

バイト7: 0Fx: 命令のMSB (内部走査リングのスキャンアウト)

【0110】3. 基本コマンドのロード: Tap状態マシンを (Exit1-IRから) シフトDRに順序づけるようスレーブに命令する。

I²C書込み開始w/R/Wビット=0、長さ=4バイト、STOP

バイト1: 02x: 4個のTCKパルス

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

バイト4: 03x: 値のTMSシーケンス (右端のLSBからMSBへ、Tap S/MをシフトDRにする)

【0111】4. 基本コマンドのロード: すべてについてTMS=0の8個のTCKパルスの2シリーズからなる2バイト・スキャンインを開始し、S/MをシフトDRに残すようスレーブに命令する。

I²C書込み開始w/R/Wビット=0、長さ=5バイト、STOP

バイト1: 06x: TCKパルス (すべての場合にTMS=0、注: 2バイトのデータがあるので、内部スレーブ論理回路が2通りの別個の8ビットSCANをキックオフするように、16個のTCKパルスが必要である。)

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

バイト4: BAx: データの第1のバイト (LSB)

バイト5: EFx: データの次のバイト

【0112】5. 基本コマンドのロード: 最後の4ビットをスキャンインすることにより、2. 5バイト・スキャンインを終了する。これは、最後のTCKを除くすべてについてTMS=0の4個のTCKパルスになる。

TMS=1の最後のTCKはTAP S/MをExit1-DRにキックする。

I²C書込み開始w/R/Wビット=0、長さ=4バイト、STOP

バイト1: C2x: 4個のTCKパルス (最後のTCKを除くすべての場合にTMS=0)

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

バイト4: BAx: スキャンイン・データの上位4ビット (右端のビット)

【0113】6. 次の基本コマンドのロード: Tap状態マシンをテスト論理リセット状態にするようスレーブに命令する。

I²C書込み開始w/R/Wビット=0、長さ=4バイト、STOP

バイト1: 03x: 5個のTCKパルス

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

バイト4: 1Fx: 右端のビットから左端までのTMS値のシーケンス (Tap状態マシンをリセットに戻すために5ビットのみ使用する)

【0114】ヌルTAPコマンド (TapCmdアドレス (11:8) = 「0001」)

TCK/TMS/TDI上でのJTAG活動は抑制される。このモードは、たとえば、新しいコマンドをIRにスキャンインするTap Cmdシーケンスの直後のIR状況情報を読み戻すためのものである。

【0115】SSRデータ (IOバッファ) へのI²C書込みは何の効果もない。SSRデータのI²C読取りは、スキャンアウトした最新のJTAG/TDOデータを返すことになる。

【0116】Tapコマンド例: 命令のスキャンイン直後のIR状況の読取り

1. 基本コマンドのロード: Tap状態マシンをまずテスト論理リセットにし、次にシフトIRにして、命令をスキャンインするようスレーブに命令する。

I²C書込み開始w/R/Wビット=0、長さ=5バイト、STOP

バイト1: 08x: 10個のTCKパルス (アドレス/コマンド・バイト後のデータがTMSシーケンスになる)

バイト2: 40x: 基本コマンド・アドレスのLSB

バイト3: 52x: 基本コマンド・アドレスのMSB

バイト4: Dfx: 値のTMSシーケンス (右端のLSBからMSBへ)

バイト5: 00x: 連続した値のTMSシーケンス (Tap SMをシフトIRにする)

【0117】2. 基本コマンドおよび命令データのロード: Tapコントローラ命令レジスタに0F 80 00 41をスキャンインするようスレーブに命令す

る。これは内部リングのスキャンアウトであり、次に Tap S/M を Exit 1-IR にする。

I²C 書込み開始 w/R/W ビット = 0、長さ = 7 バイト、STOP

バイト 1 : Dex : 32 個の TCK パルス

バイト 2 : 40x : 基本コマンド・アドレスの LSB

バイト 3 : 52x : 基本コマンド・アドレスの MSB

バイト 4 : 41x : 命令の LSB

バイト 5 : 00x : 命令の次のバイト

バイト 6 : 80x : 命令の次のバイト

バイト 7 : 0Fx : 命令の MSB (内部走査リングのスキャンアウト)

【0118】3. 基本コマンドのロード: 走査コントローラから最後にスキャンアウトされたものを読み取るようスレーブに命令する (命令レジスタ走査が最後の動作だったので、これは IR-STATUS になる)。

I²C 書込み開始 w/R/W ビット = 0、長さ = 3 バイト、STOP

バイト 1 : --x : このバイトは、このコマンドでは無指定になる

バイト 2 : 41x : 基本コマンド・アドレスの LSB

バイト 3 : 52x : 基本コマンド・アドレスの MSB

【0119】4. 基本読取りコマンド用の戻りデータの獲得: スキャンアウトされたデータ (以下のダッシュで表されるもの) を返す。

I²C 読取り再始動 w/R/W ビット = 1、長さ = 4 バイト、STOP

バイト 1 : --x : IR 状況の LSB

バイト 2 : --x : IR 状況の次のバイト

バイト 3 : --x : IR 状況の次のバイト

バイト 4 : --x : IR 状況の MSB

【0120】5. 次の基本コマンドのロード: Tap 状況マシンをテスト論理リセット状態にするようスレーブに命令する。

I²C 書込み開始 w/R/W ビット = 0、長さ = 4 バイト、STOP

バイト 1 : 03x : 5 個の TCK パルス

バイト 2 : 40x : 基本コマンド・アドレスの LSB

バイト 3 : 52x : 基本コマンド・アドレスの MSB

バイト 4 : 1Fx : 右端のビットから左端までの TMS 値のシーケンス (Tap 状態マシンをリセットに戻すために 5 ビットのみ使用する)

【0121】I²C 停止シーケンスの発行

【0122】アテンション使用可能コマンド (Tap Cmd アドレス (11:8) = 「0010」)。アクセスから提起されたアテンションを使用可能にする I²C 方法。

【0123】SSR データ (IO バッファ) への I²C 書込みにより、I²C スレーブによるアテンション検査が使用可能になる。JTAG 活動は抑制される。

【0124】ただし、このシーケンスが書き込まれると、スレーブは、アテンションが活動状態ではない場合のみ、レジスタ読取り/書込み動作を肯定応答することになることに留意されたい。アテンションが活動状態である場合、標準の I²C ack パルスにより基本 Tap コマンドのみ肯定応答される。

【0125】Tap コマンド例: 内部でのアテンション検査の再使用可能化

1. I²C 書込み開始 w/R/W ビット = 0、長さ = 4 バイト、STOP

2. バイト 1 : --x :

バイト 2 : 42x : 基本コマンド・アドレスの LSB

バイト 3 : 52x : 基本コマンド・アドレスの MSB

3. I²C 停止シーケンスの発行

【0126】アテンション使用不能コマンド (Tap Cmd アドレス (11:8) = 「0011」)。アクセスから提起されたアテンションを使用不能にする I²C 方法。

【0127】SSR データ (IO バッファ) への I²C 書込みにより、I²C スレーブによるアテンション検査が使用不能になる。

【0128】ただし、このシーケンスが書き込まれると、スレーブは、アテンションが活動状態である場合でも、基本コマンドならびにレジスタ読取り/書込みシーケンスを肯定応答することになることに留意されたい。この方法は、次にアテンションを提起させることになる活動状態のマシン検査などの発行を無効にするために使用する。これはレジスタ読取り/書込みエラーから提起されるアテンションを矯正するものではなく、レジスタ読取り/書込みエラー後に受信したデータは無視しなければならない。デフォルトでは、アテンションはマスキングされる。

【0129】Tap コマンド例: 内部でのアテンションのマスキング

1. I²C 書込み開始 w/R/W ビット = 0、長さ = 4 バイト、STOP

2. バイト 1 : --x :

バイト 2 : 43x : 基本コマンド・アドレスの LSB

バイト 3 : 52x : 基本コマンド・アドレスの MSB

3. I²C 停止シーケンスの発行

【0130】基本コマンド例: 内部での CRC 検査の使用可能化。このバイト・シーケンスにより CRC 検査がオンになる。

1. I²C 書込み開始 w/R/W ビット = 0、長さ = 3 バイト、STOP

2. バイト 1 : --x :

バイト 2 : 45x : 基本コマンド・アドレスの LSB

バイト 3 : 52x : 基本コマンド・アドレスの MSB

3. I²C 停止シーケンスの発行

【0131】注: バイト 2 : 45x および バイト 3 :

52xがスレーブに送信されるまで、CRC検査は使用可能になる。

【0132】例：CRCを使用するレジスタ読取りの実行

この例の場合、CRCの一実施態様では、8ビットのCRCアキュムレータ用として $x^8 + x^4 + x^3 + x^2 + 1$ を使用する。

【0133】CRC検査をオンにするための書込み基本デコード

1. I²C書込み開始w/R/Wビット=0、長さ=3
バイト、STOP

2. バイト1：--x：

バイト2：45x：基本コマンド・アドレスのLSB

バイト3：52x：基本コマンド・アドレスのMSB

3. I²C停止シーケンスの発行

【0134】(CRC_CHECKINGが使用可能になる。)

【0135】読み取るべきアドレスならびにスレーブ・アドレスおよび3バイト・アドレス用のCRCバイト(この例では59x)の送出

1. I²C書込み開始w/R/Wビット=0、長さ=4
バイト、STOP

2. バイト1：03x：レジスタ読取り/書込みアドレスのLSB

バイト2：00x：レジスタ読取り/書込みアドレスの中間バイト

バイト3：80x：レジスタ読取り/書込みアドレスのMSB

バイト4：59x：8個の0のシフトイン後のCRCバイト

3. I²C停止シーケンスの発行

【0136】(レジスタ読取り/書込みアドレス800003がスレーブに記憶される。)

【0137】完了まで必要なバイト数を読み取る。CRCバイトはI²Cバス上で返送されるデータについてのみ計算される。

1. I²C書込み開始w/R/Wビット=1、長さ=8
バイト、STOP

この例ではバイト1～8を読み取る(可変長)

2. I²C停止シーケンスの発行

【0138】最後の読取り動作時に計算されたCRCバイトを入手するためには、CRCバイトとしての「11」xとともに基本デコード「5247」xを伝送する必要がある(コマンドの無指定部分で「03」xが送信される場合)。

1. I²C書込み開始w/R/Wビット=0、長さ=4
バイト、STOP

2. バイト1：--x：CRC読取りの場合の無指定(この場合には03xを使用する)

バイト2：47x：読取りCRCデコード・アドレス

バイト3：52x：基本アドレスのMSB

バイト4：11x：8個の0のシフトイン後のCRCバイト

3. I²C停止シーケンスの発行

【0139】(レジスタ読取り/書込みアドレス800003がスレーブに記憶される。)

【0140】CRC読取り用の基本デコードを送出した後、実行されたレジスタ読取りに関するCRCを入手するために読取りシーケンスを実行することができる。1バイトのみ有効なので、1バイトだけを読み取るかまたは残りを無視しなければならない。マスタはこのバイトに対してno-ack(「1」b)でなければならないので、バス・エラーを引き起こさずにこの動作を実行することが可能である。

1. I²C書込み開始w/R/Wビット=1、長さ=1
バイト、STOP

この例ではバイト#1を読み取る(1バイトのみ有効である)

2. I²C停止シーケンスの発行

【0141】ただし、レジスタをゼロにするために、前に計算したバイトによりCRCバイトを送信しなければならないことに留意されたい。

【0142】このバイト・シーケンスにより、CRC検査がオフになる。

1. I²C書込み開始w/R/Wビット=0、長さ=3
バイト、STOP

2. バイト1：--x：TCKパルス

バイト2：46x：基本コマンド・アドレスのLSB

バイト3：52x：基本コマンド・アドレスのMSB

3. I²C停止シーケンスの発行

【0143】次のプログラマのためにCRC検査が使用不能になる。

【0144】まとめとして、本発明の構成に関して以下の事項を開示する。

【0145】(1)グローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置であって、前記グローバル・シリアル・バスとの間で転送すべきデータをバッファする入出力バッファ論理回路と、前記入出力バッファ論理回路に接続され、データを受信し、そのデータを前記入出力バッファ論理回路に送信するスレーブ・インタフェース論理回路と、前記入出力バッファ論理回路および前記スレーブ・インタフェース論理回路に結合され、前記入出力バッファ論理回路に合わせてデータ交換のペースを調整するスレーブ・コントローラを含む装置。

(2)前記入出力バッファと前記グローバル・シリアル・バスとの間に結合され、エラー条件を処理するエラー処理論理回路をさらに含む、上記(1)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(3)前記エラー処理論理回路が巡回冗長検査(CR

C) 計算論理回路を含む、上記(2)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(4) 前記グローバル・シリアル・バスが集積回路間(I²C)バスを含む、上記(1)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(5) 前記製造テスト・インタフェースがジョイント・テスト・アクション・グループ(JTAG)インタフェースを含み、前記スレーブ・インタフェース論理回路が前記JTAGインタフェースとの間で送受信されるJTAGフォーマット・コマンドを処理する、上記(1)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(6) 前記入出力バッファと前記グローバル・シリアル・バスとの間に結合され、エラー条件を処理するエラー処理論理回路をさらに含み、前記JTAGインタフェースが前記エラー処理論理回路にアテンション信号を供給し、前記アテンション信号がエラーを示す、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(7) 前記エラー処理論理回路が巡回冗長検査(CRC)計算論理回路を含み、前記CRC計算論理回路がCRC計算を実現するために使用可能になる、上記(6)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(8) 前記スレーブ・インタフェース論理回路が、前記アテンション信号に応答してエラー検査を使用可能および使用不能にするように構成される、上記(6)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(9) 前記スレーブ・コントローラが、前記スレーブ・インタフェース論理回路と前記JTAGインタフェースの間のデータ交換のペースを調整する、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(10) 前記スレーブ・コントローラが、前記スレーブ・インタフェース論理回路と前記JTAGインタフェースの間のデータ交換のためにデータ・ステアリングおよび経路指定を行う、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(11) 前記スレーブ・コントローラが、レジスタ読取り/書込み動作のために順序づけた一連のコマンドを前記スレーブ・インタフェース論理回路に供給する、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(12) 前記スレーブ・インタフェース論理回路が同時に発生したJTAG読取り/書込み動作を処理し、前記スレーブ・コントローラが書込み動作に続いてJTAG

命令レジスタ(IR)状況読取り動作のための読取り動作を行う、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(13) 前記スレーブ・コントローラが、開始バイト・アドレスに続いてレジスタ読取り/書込み動作のためのレジスタ読取り/書込みアドレスを供給する、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(14) 前記スレーブ・コントローラが、走査通信レジスタ読取り/書込み動作のために順序づけた一連のコマンドを前記スレーブ・インタフェース論理回路に供給する、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(15) 前記グローバル・シリアル・バスが、集積回路間(I²C)バスと、前記I²Cバスと前記入出力バッファ論理回路の間に結合された物理インタフェースとを含み、前記スレーブ・コントローラが、前記物理インタフェースと前記入出力バッファ論理回路の間のデータ交換のペースを調整する、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(16) 前記スレーブ・コントローラが、前記物理インタフェースと前記入出力バッファ論理回路の間のデータ交換のためにデータ・ステアリングを行う、上記(15)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(17) 前記スレーブ・コントローラが、レジスタ読取り動作のために前記スレーブ・インタフェース論理回路にコマンドを供給する、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(18) 前記スレーブ・コントローラが、レジスタ書込み動作のために前記スレーブ・インタフェース論理回路にコマンドを供給する、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(19) 前記スレーブ・インタフェース論理回路がテスト・クロック(TCK)生成論理回路を含む、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(20) 前記スレーブ・インタフェース論理回路が前記JTAGインタフェースからロード・コマンド信号を受信する、上記(5)に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(21) 前記JTAGインタフェースと前記スレーブ・インタフェース論理回路が内部JTAGバスによって接続され、前記内部JTAGバス上の活動を開始するために前記ロード・コマンド信号が活動化される、上記(2

0) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(22) 前記グローバル・シリアル・バスが集積回路間 (I^2C) バスを含み、エラー処理のために肯定応答信号が使用される、上記 (5) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(23) 前記スレーブ・コントローラが、1つまたは複数の基本動作を実行するために前記スレーブ・インタフェース論理回路にコマンドを供給する、上記 (5) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(24) 前記基本動作が、前記 JTAG インタフェースへの制御、命令、データ、状況交換のうちの少なくとも1つを選択する、上記 (5) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(25) 前記 JTAG インタフェースが走査通信即時読取り論理回路を含む、上記 (5) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(26) 前記走査通信即時読取り論理回路が、ターゲット・レジスタ内容を取り出して走査通信データ・レジスタにロードし、前記走査通信データ・レジスタの前記内容を直ちに読み出すために1つのステップを実行する、上記 (5) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(27) 前記スレーブ・コントローラが、前記入出力バッファ論理回路とのデータ交換のためにデータ・ステアリングおよび経路指定を行う、上記 (1) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための装置。

(28) グローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法であって、物理インタフェースと内部バスの上にスレーブ状態マシンを設け、前記物理インタフェースが前記グローバル・シリアル・バスに接続され、前記内部バスが前記製造テスト・インタフェースに接続されるステップと、前記スレーブ状態マシンを使用するステップであって、前記グローバル・シリアル・バスとの間で転送すべきデータをバッファするステップと、前記製造テスト・インタフェースと前記グローバル・シリアル・バスとの間のデータ交換のペースを調整するステップとを実行するためのステップとを含む方法。

(29) 前記スレーブ状態マシンが、エラー条件を処理するステップをさらに実行する、上記 (28) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

(30) 前記スレーブ状態マシンが、走査通信読取り動作のためのコマンドを供給するステップをさらに実行す

る、上記 (28) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

(31) 前記スレーブ状態マシンが、走査通信書込み動作のためのコマンドを供給するステップをさらに実行する、上記 (28) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

(32) 前記スレーブ状態マシンが、制御読取り動作のためのコマンドを供給するステップをさらに実行する、上記 (28) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

(33) 前記スレーブ状態マシンが、制御書込み動作のためのコマンドを供給するステップをさらに実行する、上記 (28) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

(34) 前記スレーブ状態マシンが、基本書込み動作のためのコマンドを供給するステップをさらに実行する、上記 (28) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

(35) 前記スレーブ状態マシンが、基本読取り動作のためのコマンドを供給するステップをさらに実行する、上記 (28) に記載のグローバル・シリアル・バスに製造テスト・インタフェースを接続するための方法。

(36) グローバル・シリアル・バスに製造テスト・インタフェースを接続するための集積回路デバイスであって、前記集積回路デバイスが、前記グローバル・シリアル・バスとの間で転送すべきデータをバッファする入出力バッファ論理回路と、前記入出力バッファ論理回路に接続され、データを受信し、そのデータを前記入出力バッファ論理回路に送信するスレーブ・インタフェース論理回路と、前記入出力バッファ論理回路および前記スレーブ・インタフェース論理回路に結合され、前記入出力バッファ論理回路に合わせてデータ交換のペースを調整するスレーブ・コントローラとを含む集積回路デバイス。

(37) 前記入出力バッファ論理回路と前記スレーブ・インタフェース論理回路の間に結合されたエラー処理論理回路をさらに含む、上記 (36) に記載の集積回路デバイス。

【図面の簡単な説明】

【図1】 好ましい実施の形態による製造テスト・インタフェースおよびグローバル集積回路間 (I^2C) バス相互接続装置のブロック図表現である。

【図2】 好ましい実施の形態による図1の製造テスト・インタフェースおよびグローバル集積回路間 (I^2C) バス相互接続装置の I^2C スレーブ状態マシンのスレーブ・データおよび制御レジスタを示すブロック図表現である。

【図3】 好ましい実施の形態による図1の製造テスト・インタフェースおよびグローバル集積回路間 (I^2C) バス相互接続装置の I^2C スレーブ状態マシンのスレー

ブ I O バッファ論理回路を示すブロック図表現である。

【図 4】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンのスレーブ・テスト・アクセス・ポート (TAP) インタフェースを示すブロック図表現である。

【図 5】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンのスレーブ・コントローラを示すブロック図表現である。

【図 6】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の物理インタフェースと I²C スレーブ状態マシンの I O バッファ論理回路とのデータ・ペーシングを示すブロック図表現である。

【図 7】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンのスレーブ・テスト・アクセス・ポート (TAP) インタフェースと I²C スレーブ状態マシンの I O バッファ論理回路とのデータ・ペーシングを示すブロック図表現である。

【図 8】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンのスレーブ・エラー処理論理回路を示すブロック図表現である。

【図 9】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンの複数の状態を示すブロック図表現である。

【図 10】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンの基本書込み制御フローを示す図である。

【図 11】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンの基本読取り制御フローを示す図である。

【図 12】好ましい実施の形態による図 1 の製造テスト

・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンのレジスタ書込み制御フローを示す図である。

【図 13】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンのレジスタ読取り制御フローを示す図である。

【図 14】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンの制御書込み制御フローを示す図である。

【図 15】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置の I²C スレーブ状態マシンの制御読取り制御フローを示す図である。

【図 16】好ましい実施の形態による図 1 の製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置のテスト・アクセス・ポート (TAP) および境界走査アーキテクチャであるアクセスを示すブロック図表現である。

【符号の説明】

100 製造テスト・インタフェースおよびグローバル集積回路間 (I²C) バス相互接続装置

102 I²C 状態マシンおよびトランスレータ

104 I²C 物理インタフェース

106 I²C バス

108 マルチプレクサ (MUX)

110 内部 JTAG バス

112 外部 JTAG バス

113 制御入力

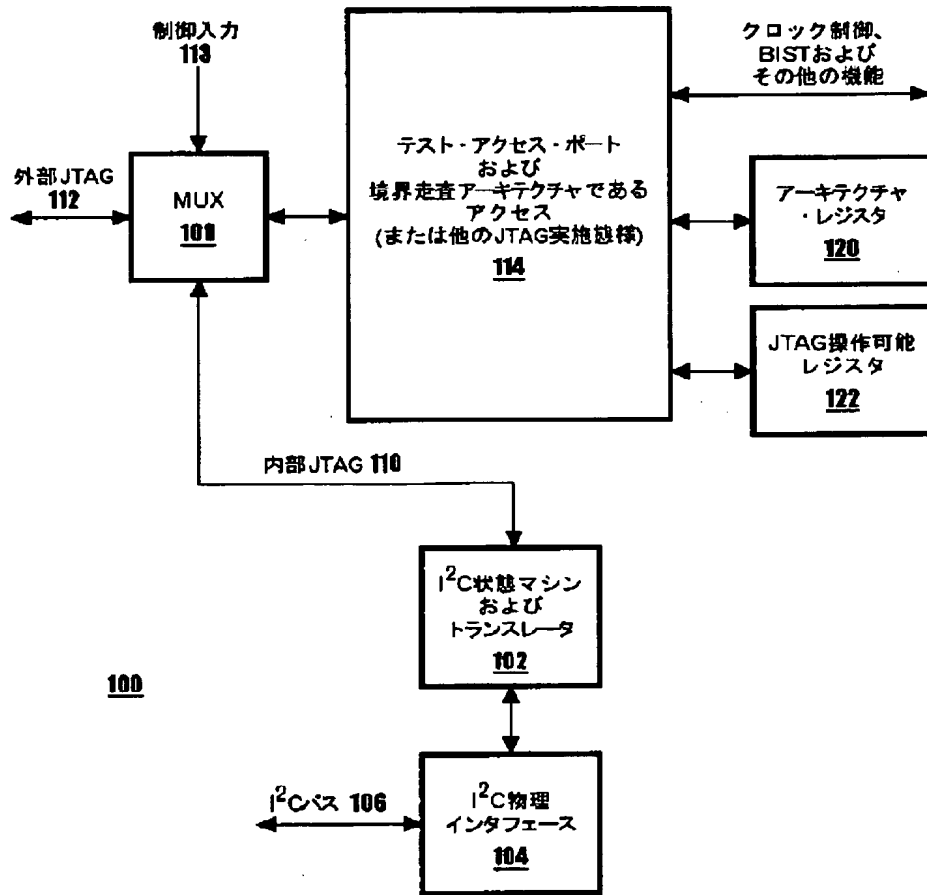
114 テスト・アクセス・ポートおよび境界走査アーキテクチャであるアクセス (または他の JTAG 実施態様)

118 クロック制御、組込み自己試験 (BIST) およびその他の機能

120 アーキテクチャ・レジスタ

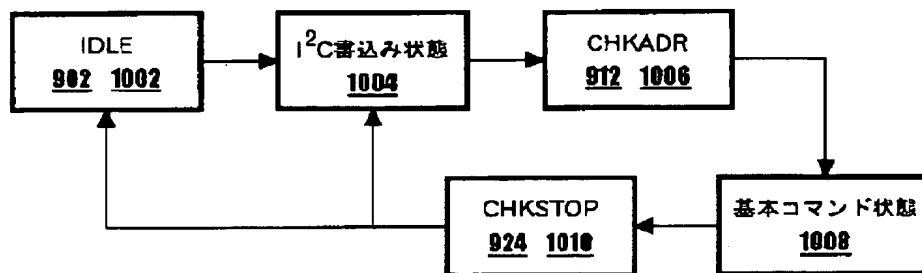
122 JTAG 走査可能レジスタ

【図1】

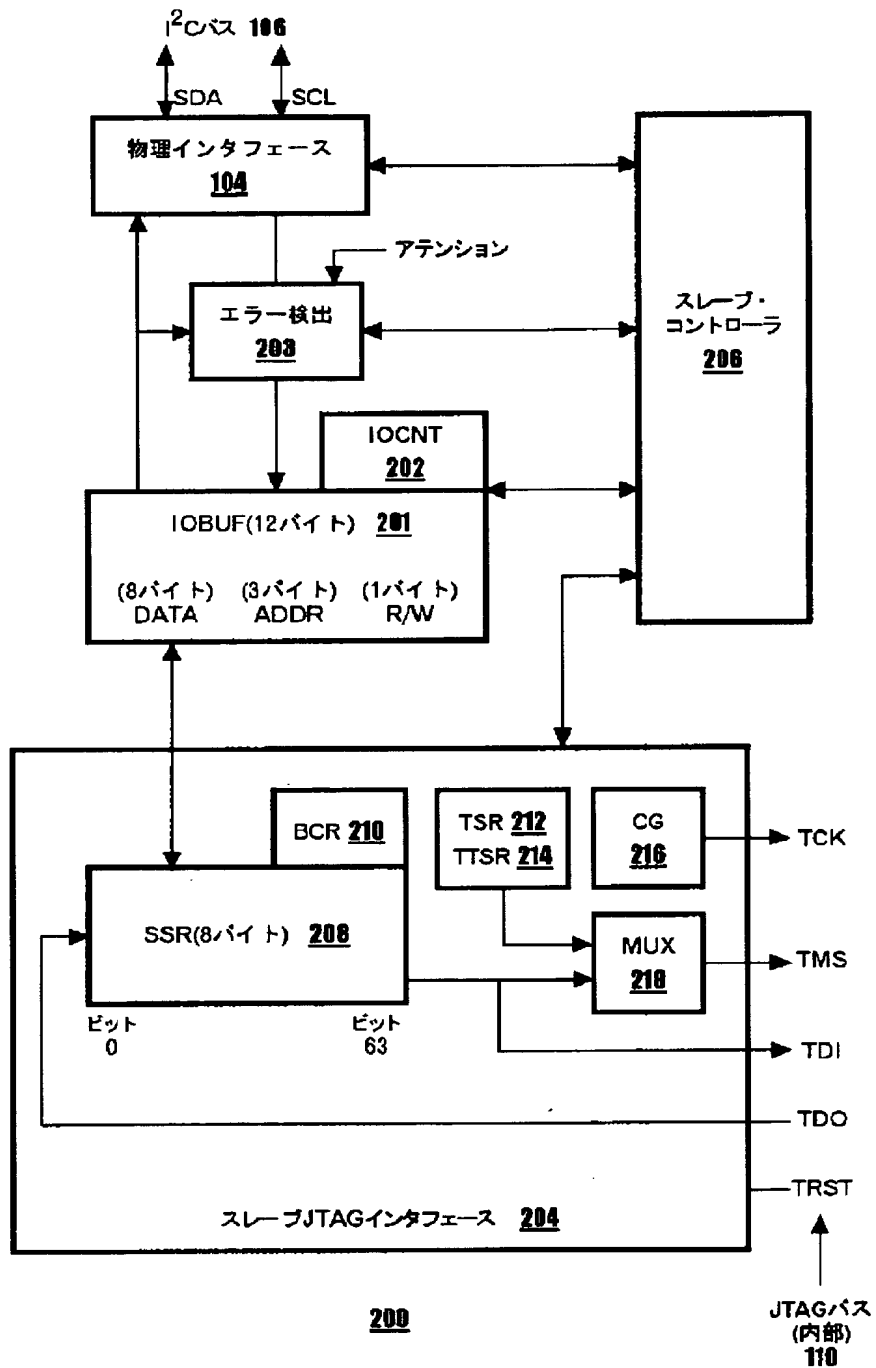


【図10】

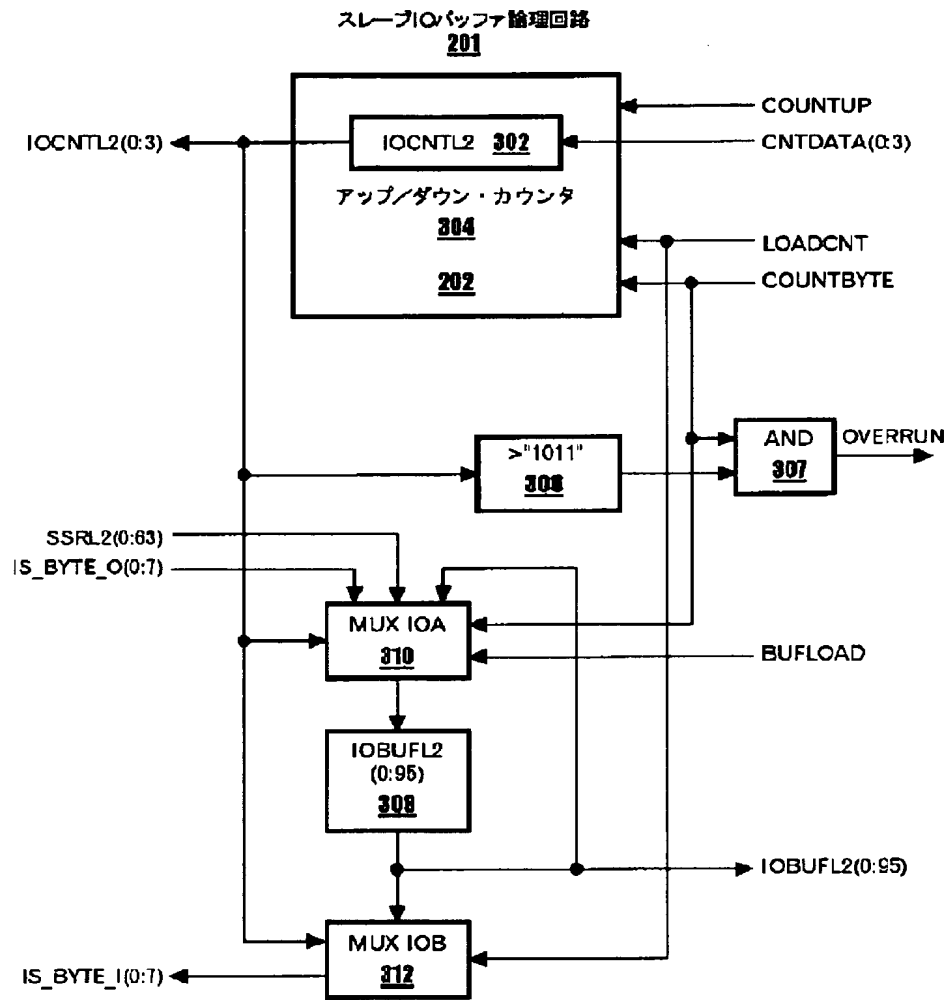
基本書込み制御フロー



【図 2】

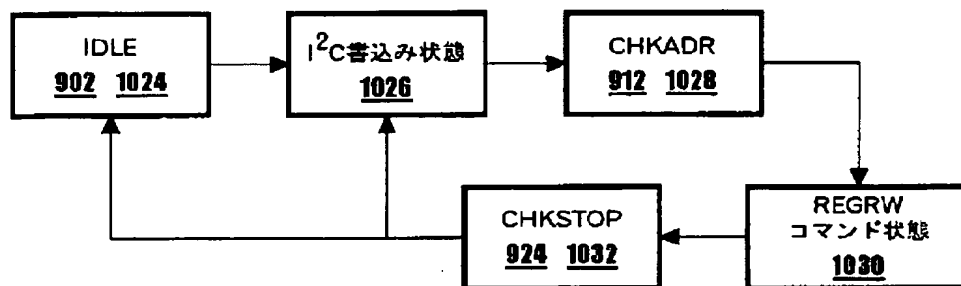


【図 3】

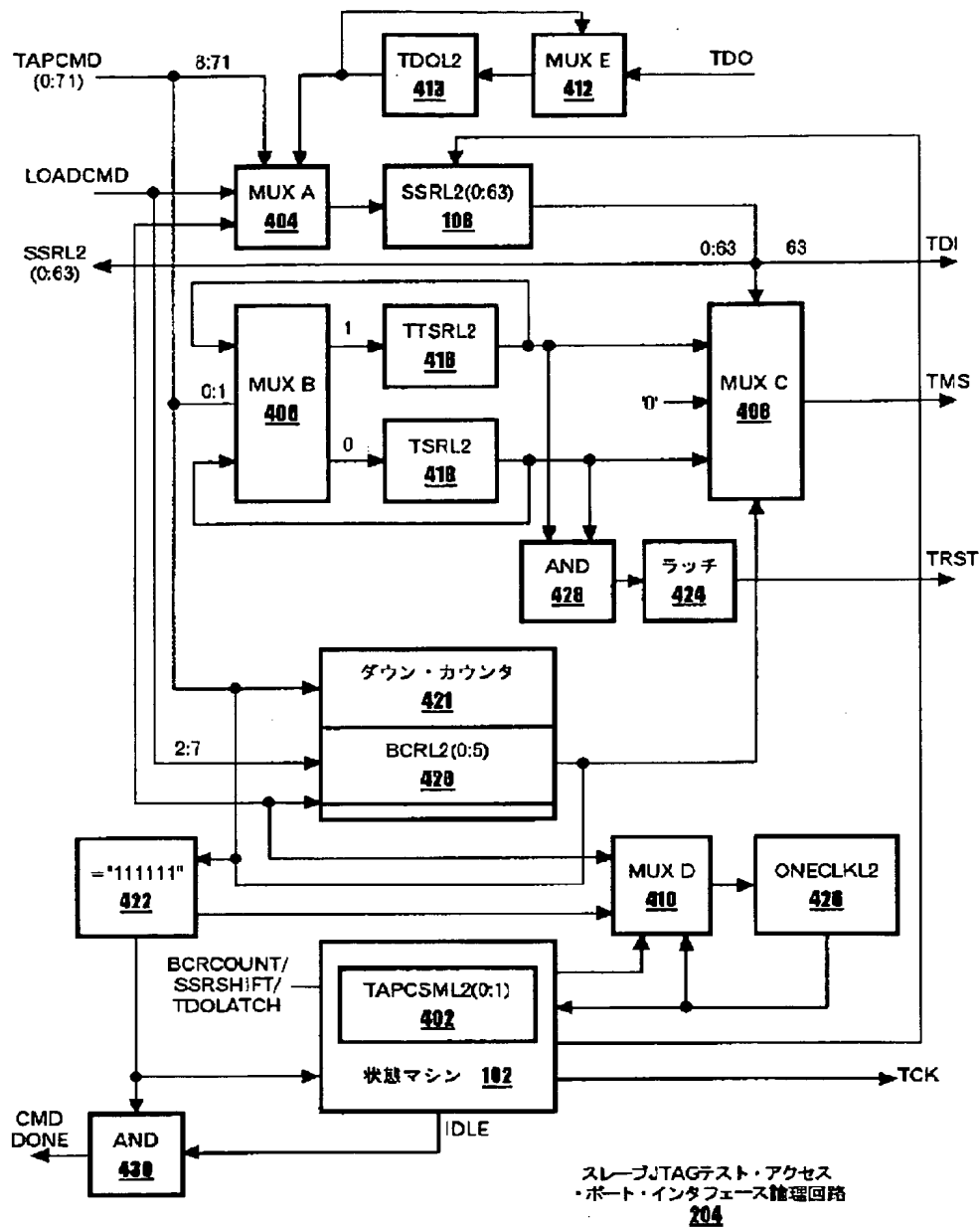


【図 12】

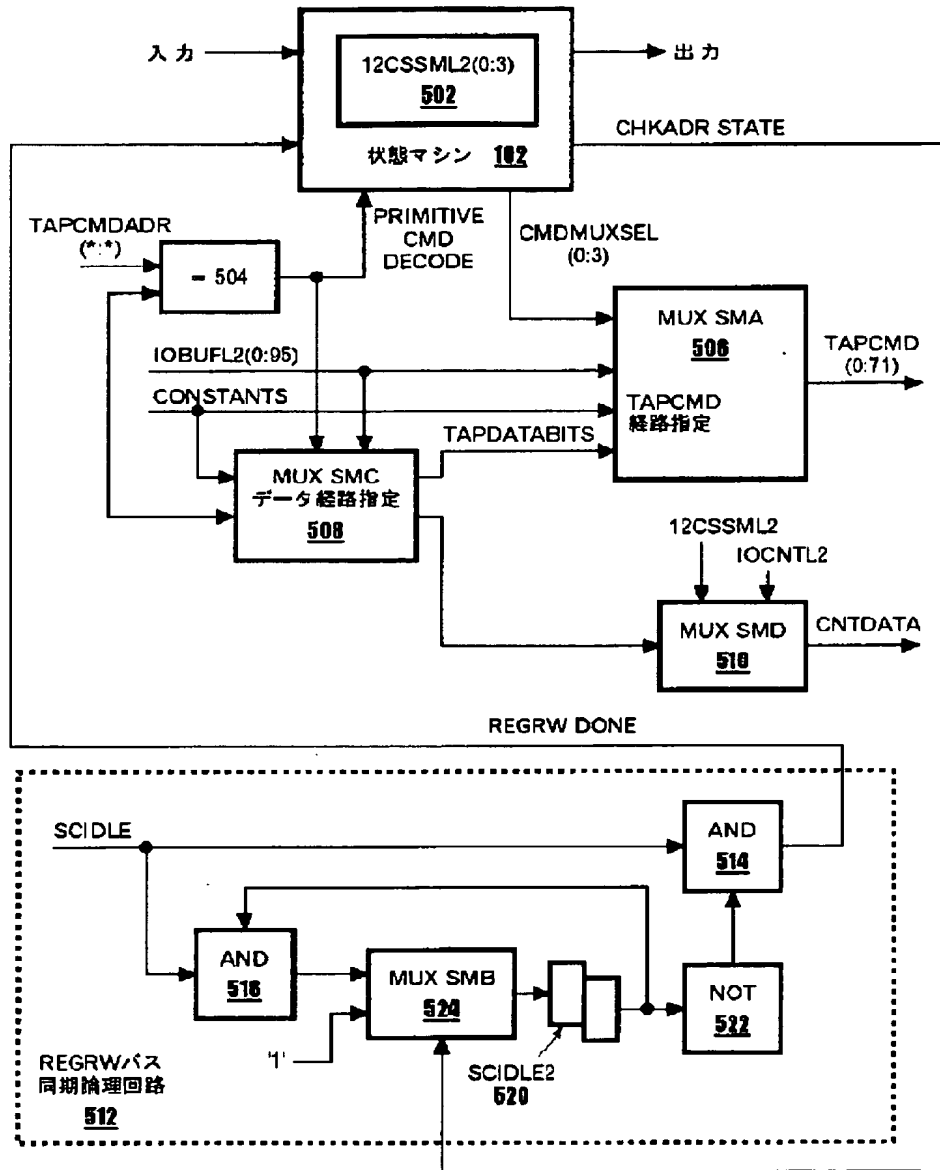
レジスタ書き込み制御フロー



【図 4】

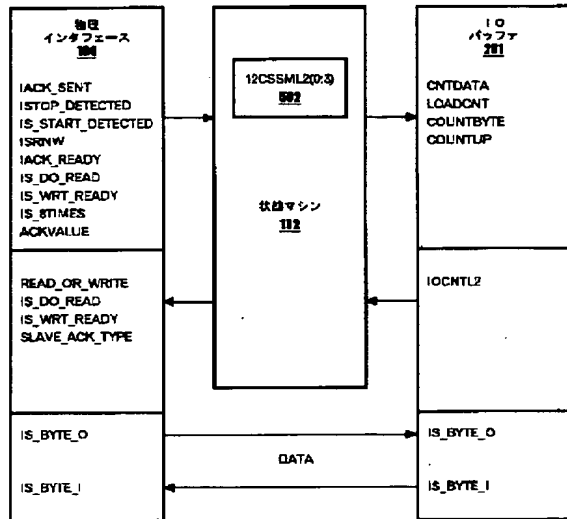


【図 5】

スレーブ・コントローラ論理回路
206

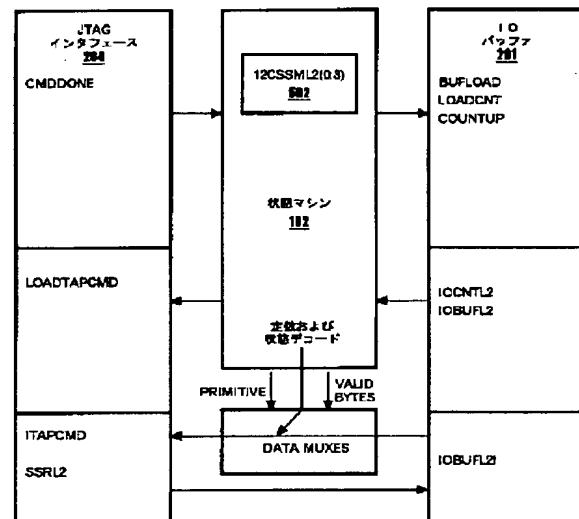
【図6】

I/Oバッファ201と物理インタフェース104とのデータ・ペーシング

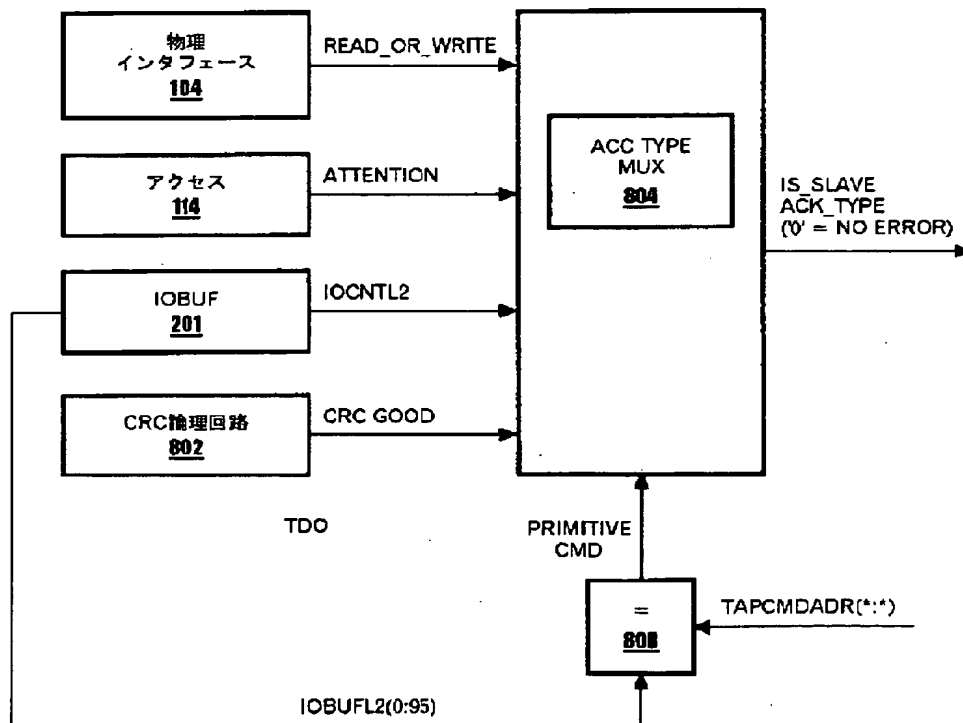


【図7】

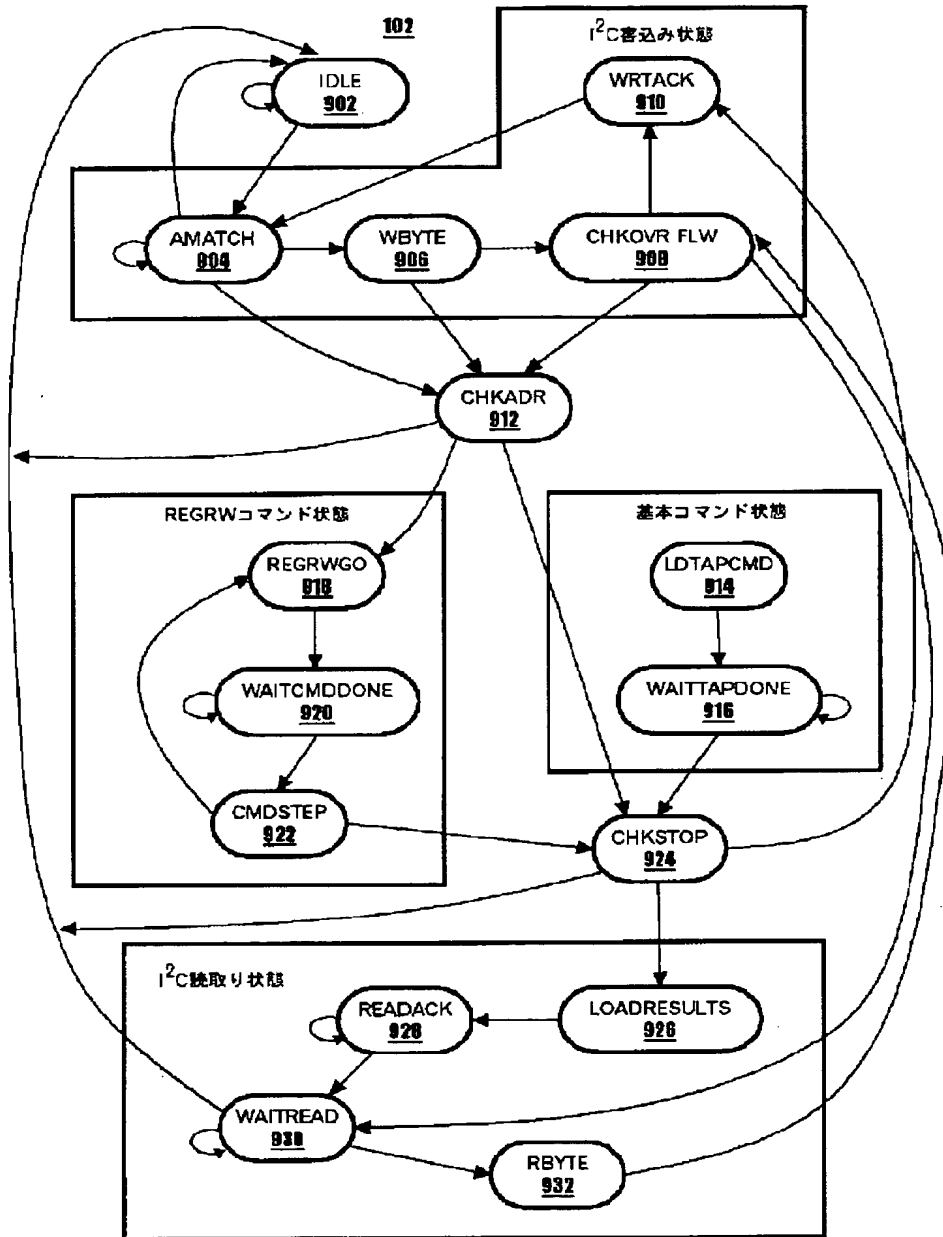
I/Oバッファ201とJTAGインタフェース204とのデータ・ペーシング



【図8】

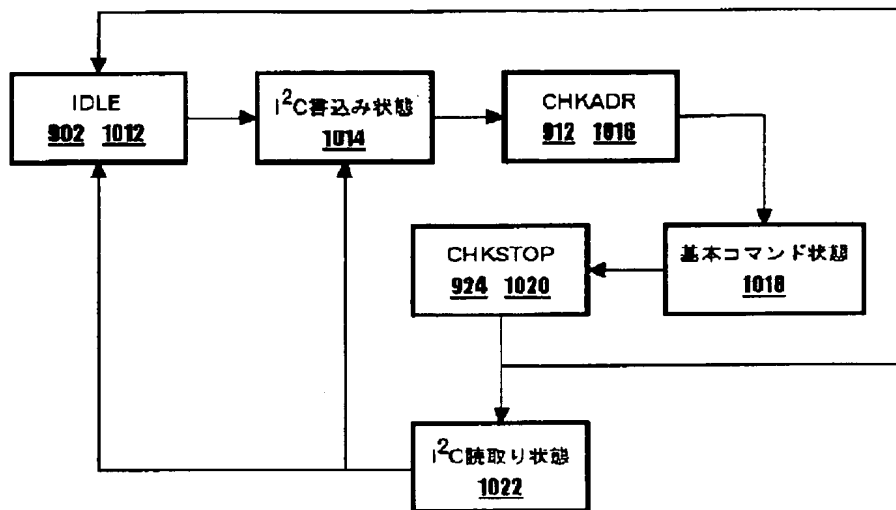
エラー処理論理回路
203

【図 9】



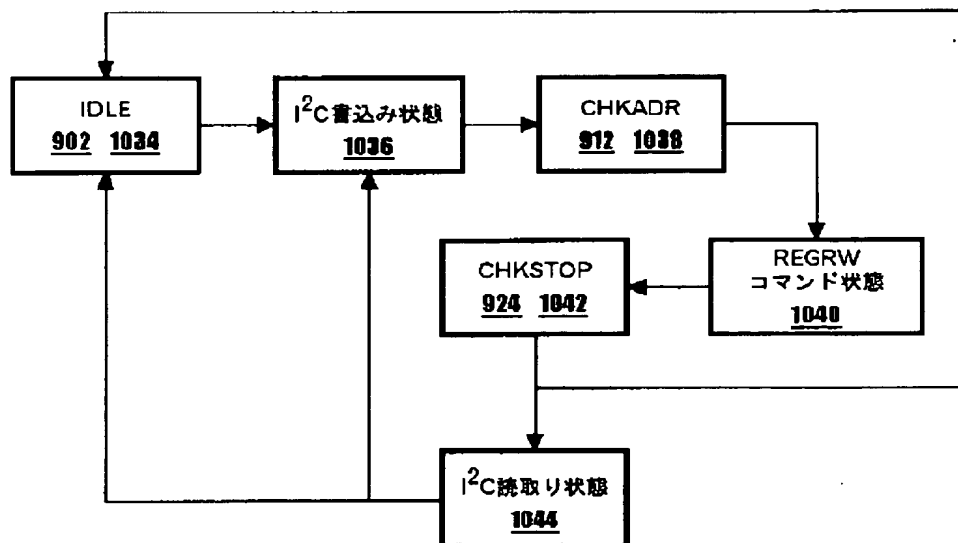
【図 11】

基本読取り制御フロー



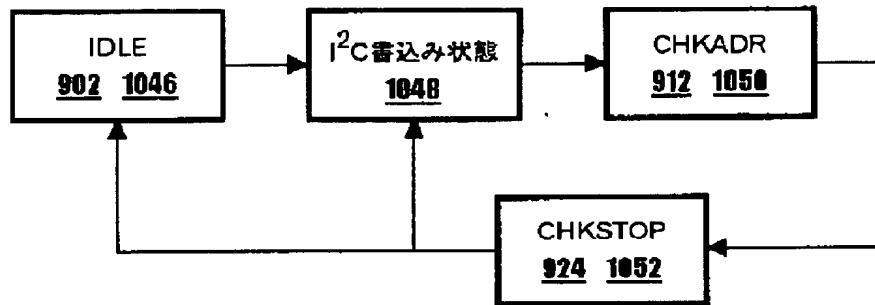
【図 13】

レジスタ読取り制御フロー



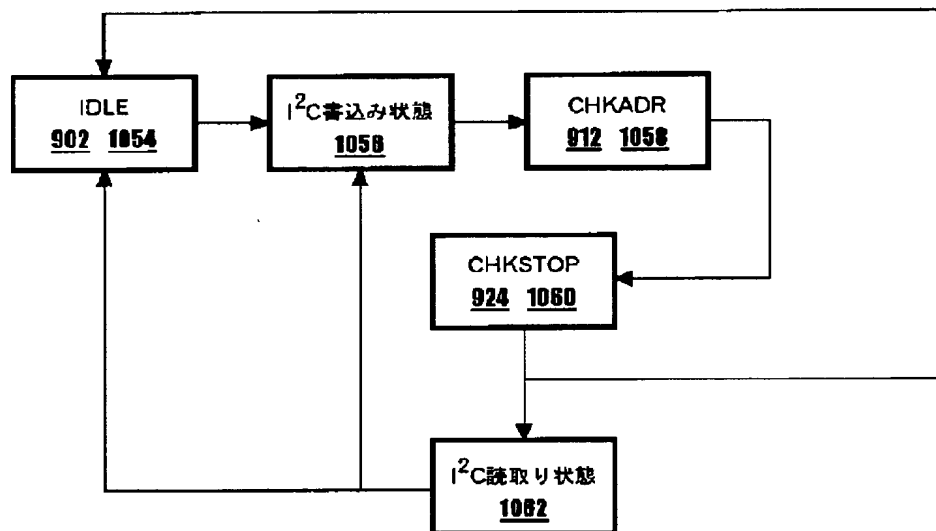
【図 14】

制御書き込み制御フロー

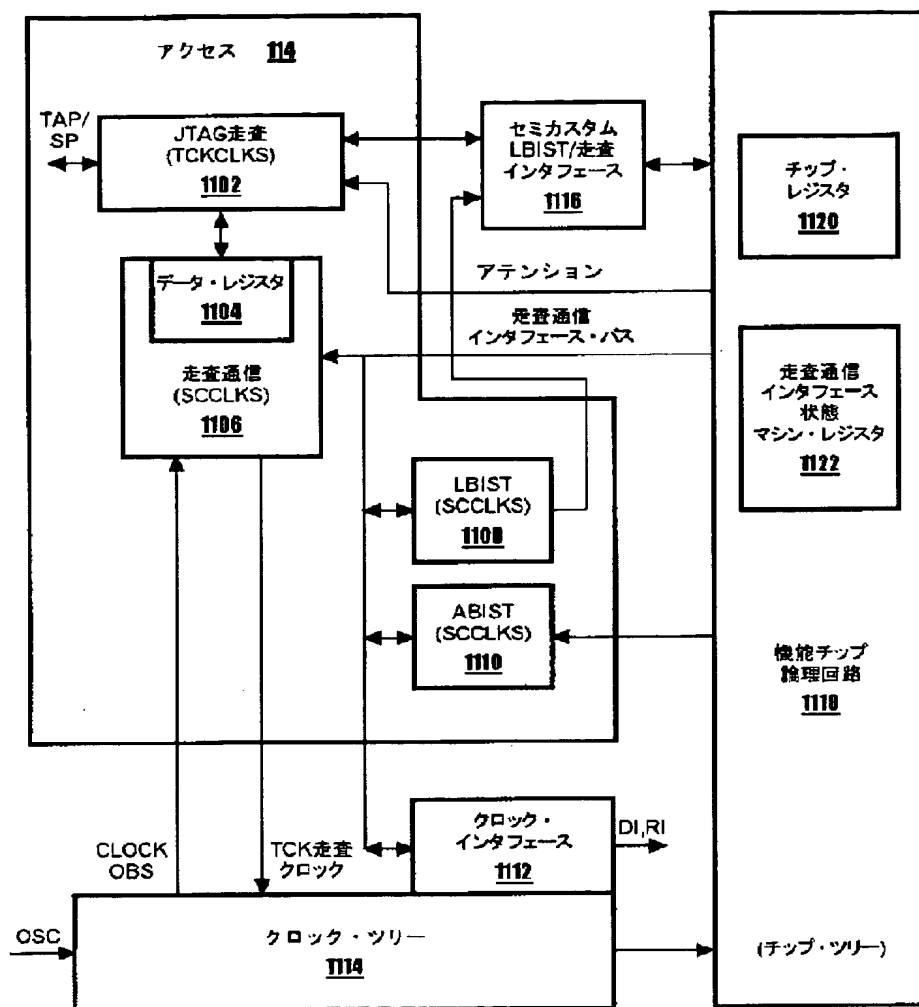


【図 15】

制御読取り制御フロー



【図16】



フロントページの続き

(72)発明者 ガイ・リチャード・クリエ
アメリカ合衆国55906 ミネソタ州ロチェ
スター テラス・レーン ノースイースト
1711

(72)発明者 ジェームズ・スコート・ハーヴランド
アメリカ合衆国55901 ミネソタ州ロチェ
スター フォーティーンズ・ストリート・
ノース・ウェスト620

(72)発明者 シャロン・デノス・ヴィンセント
アメリカ合衆国55906 ミネソタ州ロチェ
スター ブリストル・レーン ノースイ
ースト801

(72)発明者 ポール・レオナード・ウルトゲン
アメリカ合衆国55944 ミネソタ州カッソ
ン フィフス・ストリート ノースウェス
ト1202